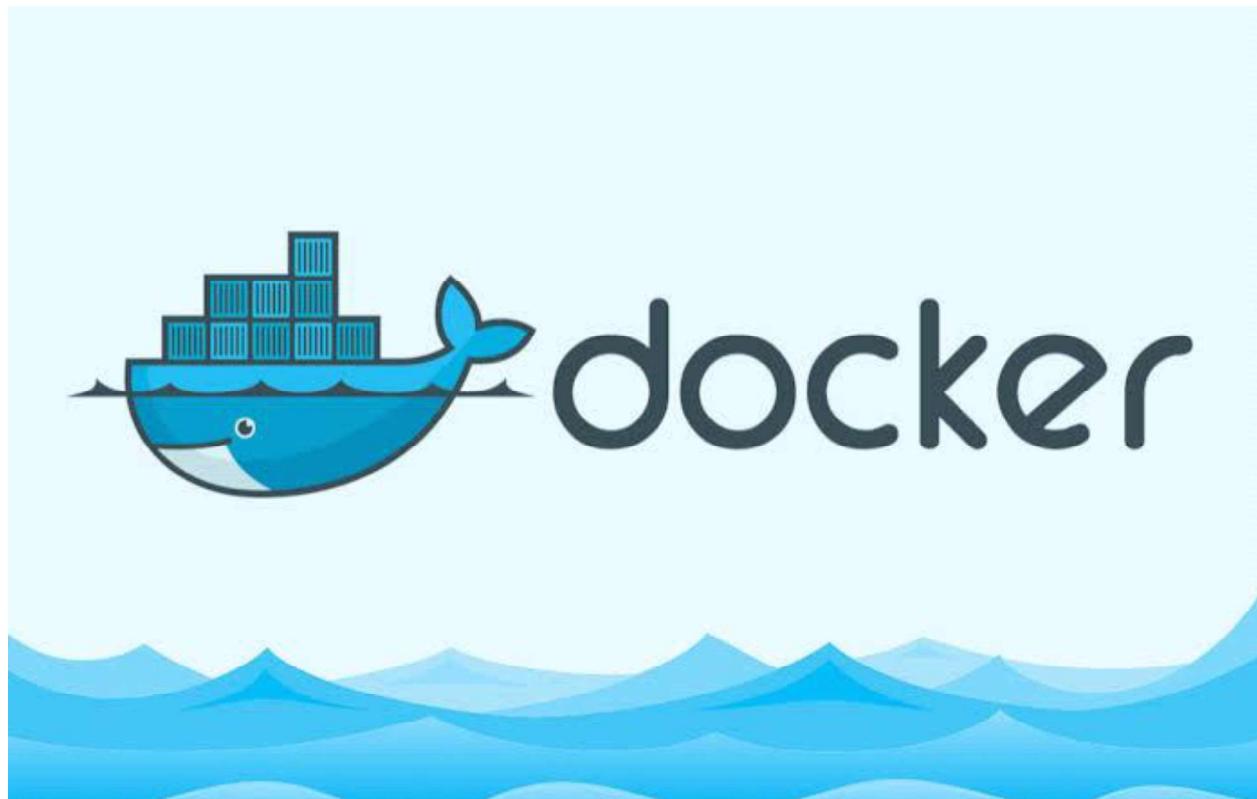


Launching a Docker Infrastructure from Zero



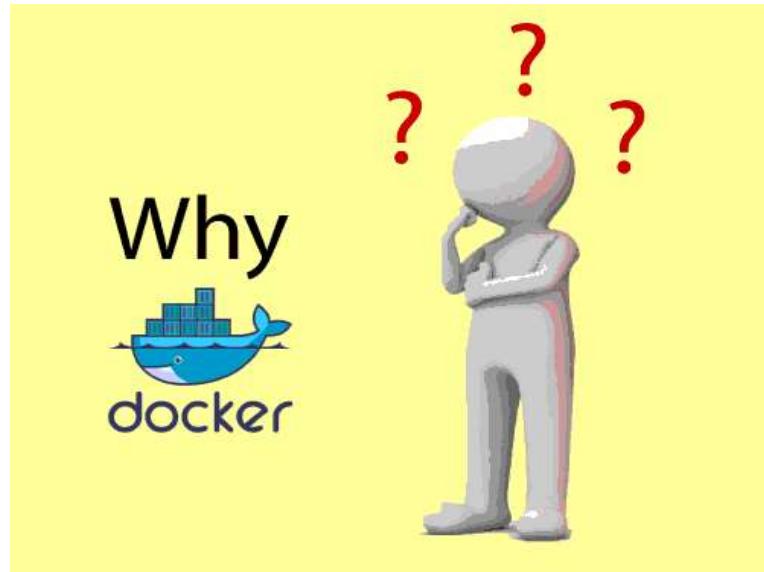
We will cover the following –

1. Introduction to Docker
2. Why Docker?
3. Advantages of Docker
4. Disadvantages of Docker
5. About Bare Metal, Virtual Machines and Containers
6. Detailed steps for launching Docker Infrastructure
 - a. Prerequisite – complete Redhat 8 installation on windows pc
 - b. Configuring yum in Red Hat
 - c. Installing docker
 - d. Now we will install docker-compose
 - e. Infrastructure for owncloud
 - f. Infrastructure for drupal

Introduction to Docker

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.

Why Docker?



Docker is designed to benefit both the Developer and System Administrator. There are the following reasons to use

1. Docker allows us to easily install and run software without worrying about setup or dependencies.
2. Developers use Docker to eliminate machine problems, i.e. "but code is worked on my laptop." when working on code together with co-workers.
3. Operators use Docker to run and manage apps in isolated containers for better compute density.
4. Enterprises use Docker to securely built agile software delivery pipelines to ship new application features faster and more securely.
5. Since docker is not only used for the deployment, but it is also a great platform for development, that's why we can efficiently increase our customer's satisfaction.

Advantages of Docker

There are the following advantages of Docker –

1. It runs the container in seconds instead of minutes.
2. It provides lightweight virtualization.
3. It does not require full operating system to run applications.
4. It uses less memory.
5. Return on Investment and Cost Saving.
6. Standardization and Productivity.
7. It eliminates the “it works on my machine” problem once and for all.
8. Docker enables you to build a container image and use that same image across every step of the deployment process.
9. Docker makes sure each container has its own resources that are isolated from other containers.
10. The last of these benefits of using docker is security

Disadvantages of Docker

There are the following disadvantages of Docker -

1. It increases complexity due to an additional layer.
2. In Docker, it is difficult to manage large amount of containers.
3. Docker is not a good solution for applications that require rich graphical interface.
4. Docker provides cross-platform compatibility means if an application is designed to run in a Docker container on Windows, then it can't run on Linux or vice versa.

Some key concepts in Docker

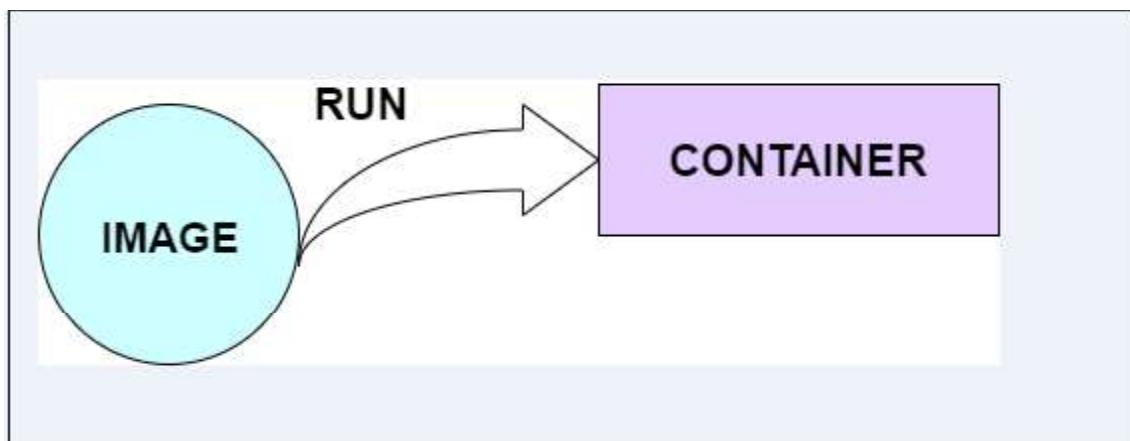
Docker Images

Docker images are the read-only binary templates used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses public container registry to share container images within the whole world. Metadata is also used by docker images to describe the container's abilities.

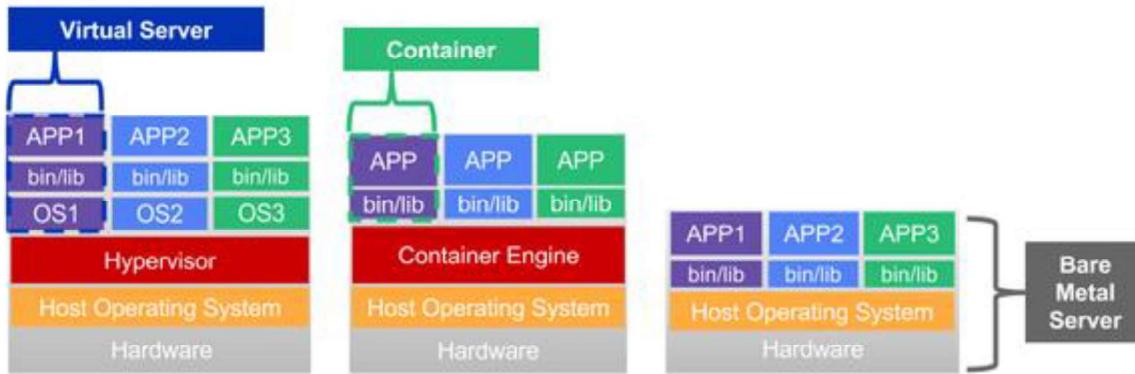
Docker Containers

Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources.

In other words, we can say that the image is a template, and the container is a copy of that template.



About Bare Metal, Virtual Machines and Containers



Bare Metal

In computer science, bare machine (or bare metal) refers to a computer executing instructions directly on logic hardware without an intervening operating system.

Virtual Machine

A virtual machine (VM) is a software program or operating system that not only exhibits the behavior of a separate computer, but is also capable of performing tasks such as running applications and programs like a separate computer.

Containers

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

Detailed steps for launching Docker Infrastructure

1

1. Here we will install VM on top of windows and on top of VM we will install a complete Red Hat 8

1. Prerequisite

You should have windows 8 / 10 installed with at-least 4 GB ram, i5 and 500 GB space.

You should have RedHat 8 .iso image (6.7 GB) and Oracle VM Virtual Box software (108 MB)

For RedHat 8 – Go to - <https://developers.redhat.com/products/rhel/download>, then press on **View Older Downloads** and click on 8.0.0 RHEL8 x86_64 DVD ISO. Create account or login using Google account and fill the required * details, the image will be downloaded

Version	Release Date	Description	Download
7.8.0	2020-03-31	RHEL x86_64	DVD ISO (4 GB) Boot ISO (611 MB)
8.2.0 Beta	2020-01-21	Boot iso	x86_64 (622 MB) aarch64 (553 MB)
		DVD iso	x86_64 (8 GB) aarch64 (6 GB)
8.1.0	2019-11-05	DVD iso	RHEL8 x86_64 (7 GB)
		Boot iso	RHEL8 x86_64 (564 MB)
		DVD iso	RHEL8 aarch64 (5 GB)
		Boot iso	RHEL8 aarch64 (526 MB)
7.7.0	2019-08-06	RHEL x86_64	DVD ISO (4 GB) Boot ISO (593 MB)
8.0.0	2019-05-07	RHEL8 x86_64	DVD ISO (7 GB) Boot ISO (533 MB)
		RHEL8 aarch64	DVD ISO (5 GB) Boot ISO (496 MB)
7.6.0	2018-10-31	RHEL x86_64	DVD ISO (4 GB) Boot ISO (545 MB)
		RHEL aarch64	DVD ISO (3 GB) Boot ISO (495 MB)

For Oracle VM Virtual Box press <https://download.virtualbox.org/virtualbox/6.1.2/VirtualBox-6.1.2-135663-Win.exe> or download from internet and .exe file will get downloaded.

Then you will have the following with you -

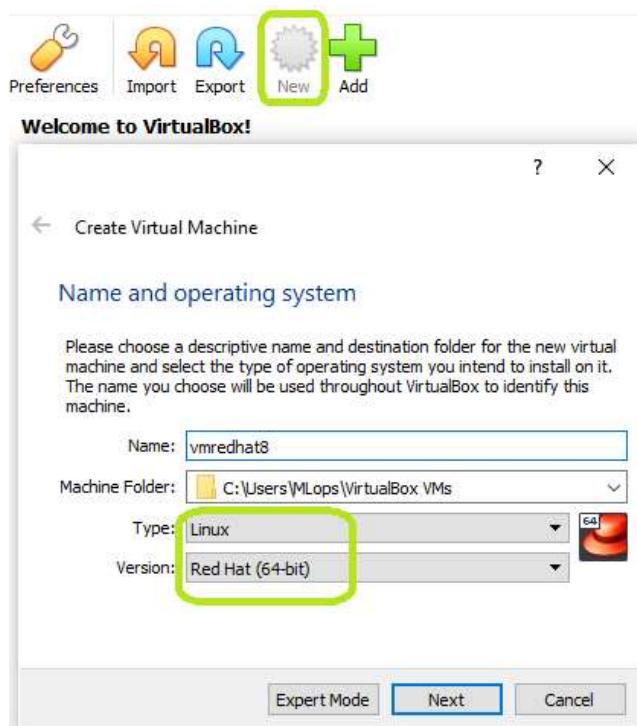
Name	Date modified	Type	Size
rhel-8.0-x86_64-dvd.iso	10-02-2020 19:37	Disc Image File	69,36,576 KB
VirtualBox-6.1.2-135663-Win.exe	10-02-2020 18:45	Application	1,10,102 KB

Now install VirtualBox as per default settings, now search for VM in windows search bar and run it... now we will install redhat 8 on top of Oracle Virtual box

Steps are as follows –

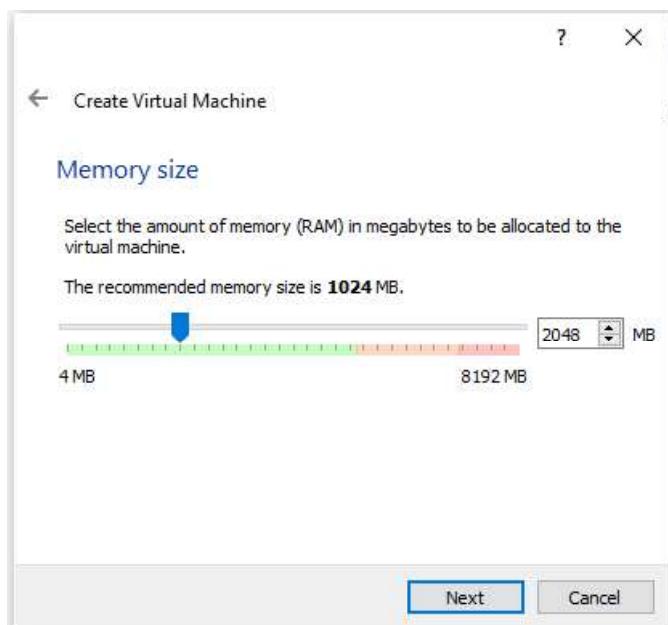
Step 1 –

Press on “New” and select “Type” as “Linux”, “Version” as “Red Hat (64-bit)” and give some name and do “Next”



Step 2 –

Enter “2048” as ram and do “Next”



Step 3 –

Say “Create”, then “Next”, then again “Next”, enter “80.00” and say “Create”, and this is how the virtual hardware is created now.

The screenshot shows the process of creating a new virtual machine in VirtualBox. It consists of two windows side-by-side:

- Left Window (Create Virtual Machine - Hard disk):** This window is for configuring the hard disk settings. It includes:
 - A note about adding a virtual hard disk to the new machine.
 - A note about skipping storage setup if changes will be made later.
 - The recommended size is **8.00 GB**.
 - Three options:
 - Do not add a virtual hard disk
 - Create a virtual hard disk now
 - Use an existing virtual hard disk file
 - A dropdown menu showing "redhat8.vdi (Normal, 80.00 GB)".
 - Buttons: **Create** (highlighted), **Cancel**.

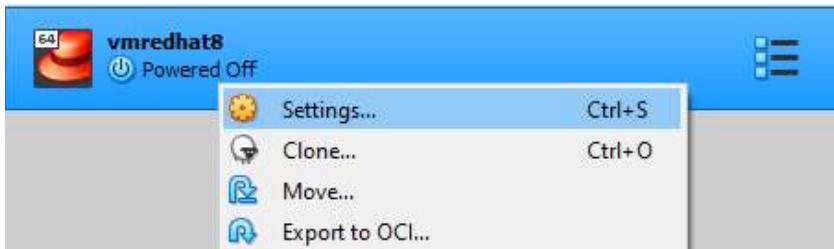
Right Window (Create Virtual Hard Disk - Hard disk file type): This window is for selecting the file type. It includes:
 - A note about choosing the type of file for the new virtual hard disk.
 - Three radio buttons:
 - VDI (VirtualBox Disk Image) (highlighted)
 - VHD (Virtual Hard Disk)
 - VMDK (Virtual Machine Disk)
 - Buttons: **Expert Mode**, **Next** (highlighted), **Cancel**.

Bottom Left Window (Create Virtual Hard Disk - Storage on physical hard disk): This window is for choosing storage allocation. It includes:
 - A note about whether the new virtual hard disk file should grow dynamically or be fixed in size.
 - A note about dynamically allocated disks using space on the physical disk as it fills up.
 - A note about fixed size disks being faster but taking longer to create.
 - Two radio buttons:
 - Dynamically allocated
 - Fixed size
 - Buttons: **Next**, **Cancel**.

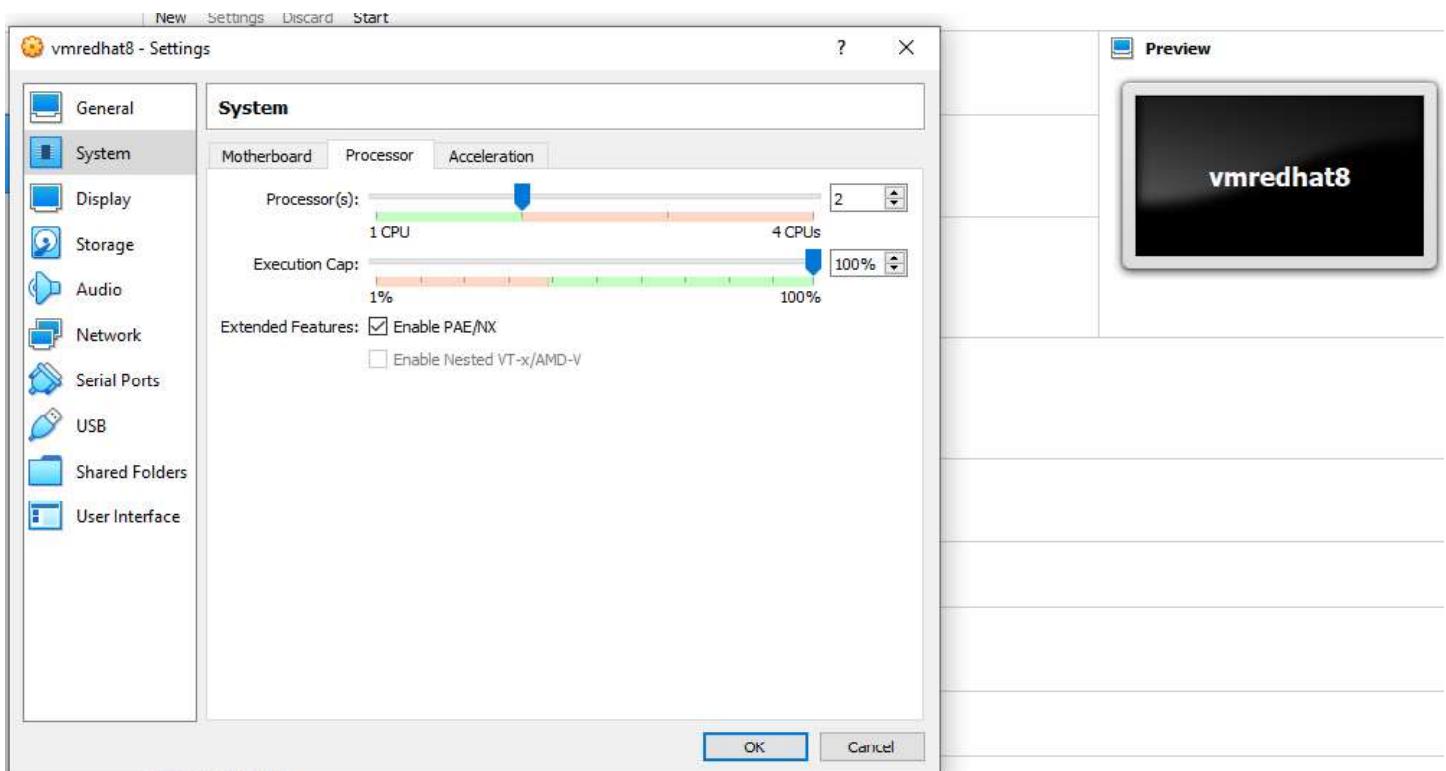
Bottom Right Window (Create Virtual Hard Disk - File location and size): This window is for specifying the file location and size. It includes:
 - A note about the file location and size.
 - A text input field containing "C:\Users\MLops\VirtualBox VMs\vmredhat8\vmredhat8.vdi".
 - A slider for selecting the size in megabytes, currently set at 80.00 GB.
 - Buttons: **Create**, **Cancel**.

Step 4 –

Right click and go to “Settings...”

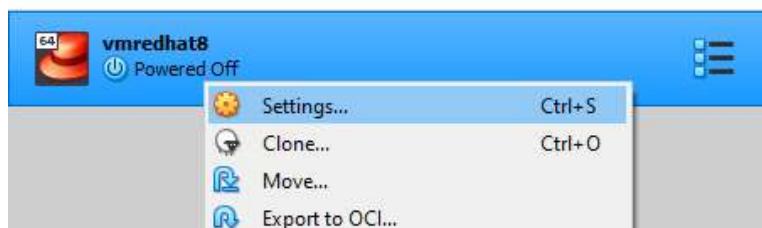


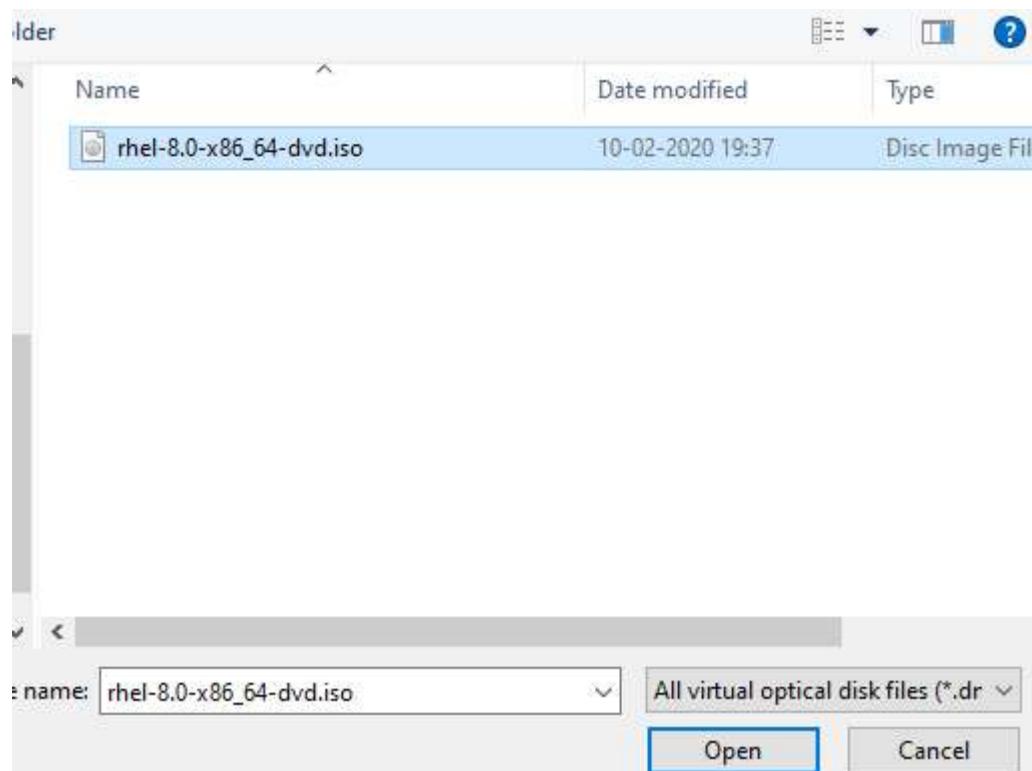
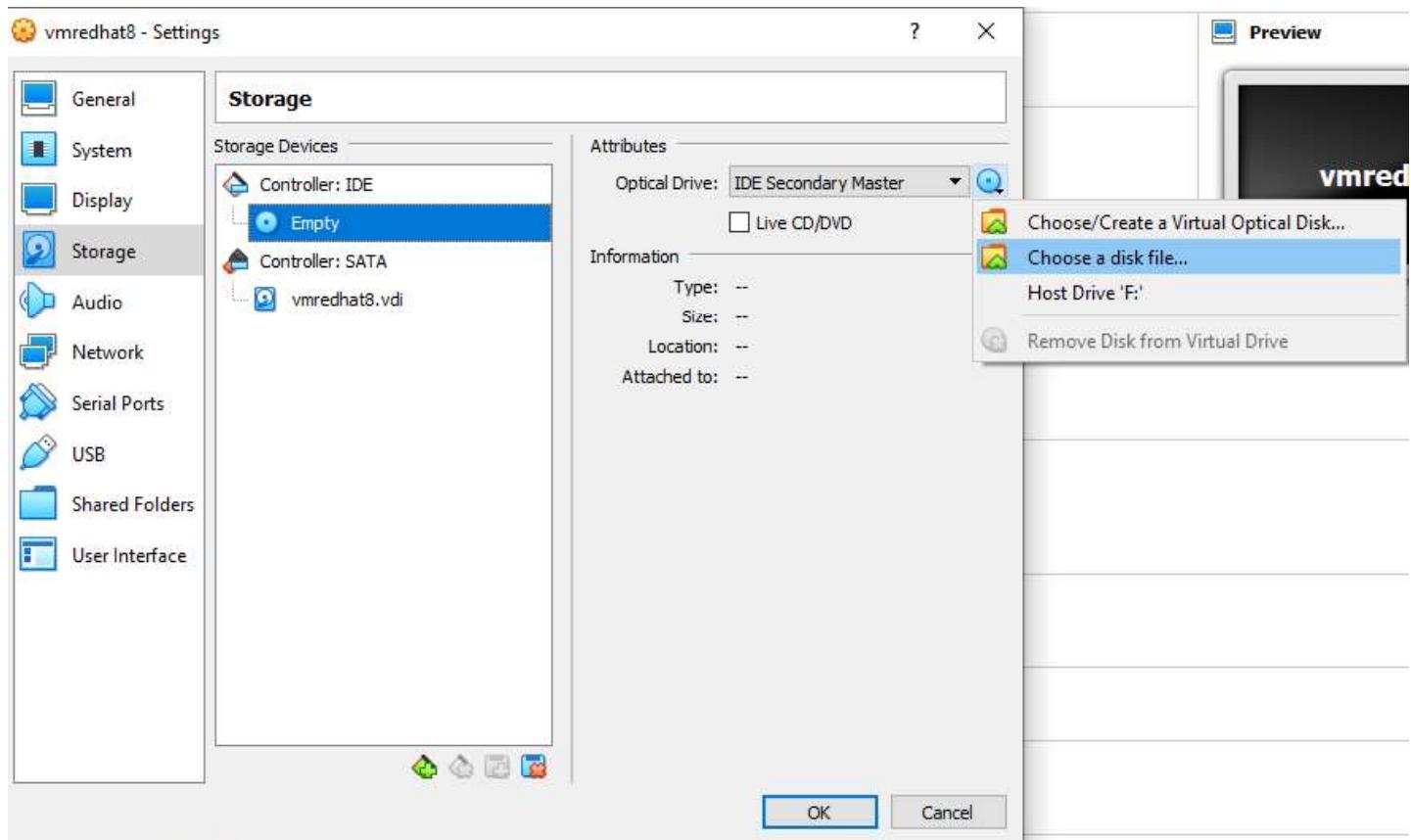
Go to “System”, then go to “Processor” and make Processor count as “2” and say “Ok”



Step 5 –

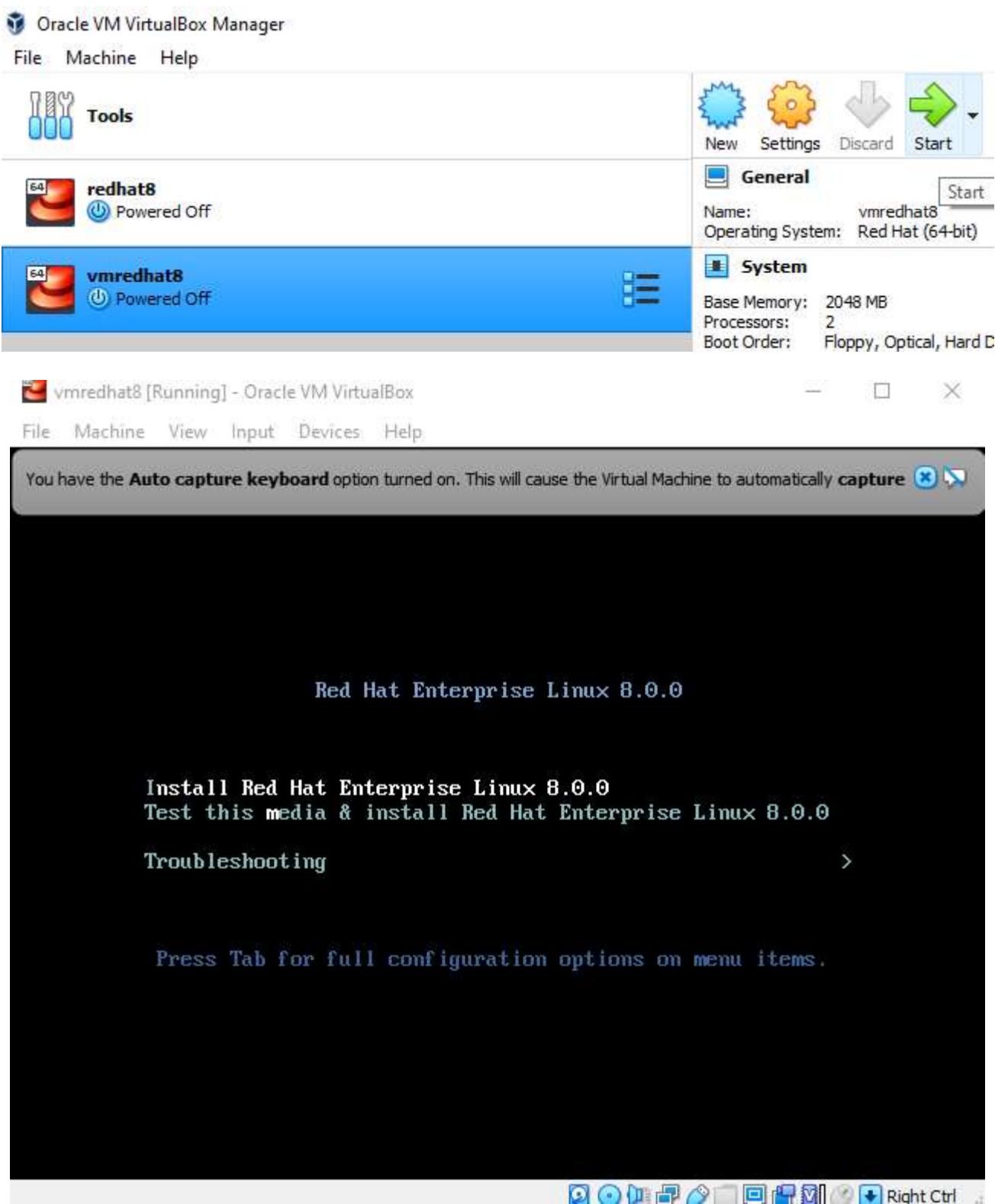
Again right click and go to “Settings”, then go to “Storage”, click on “Empty”, then click disk icon, press “Choose a disk file...”, browse for .iso file(6.7GB) of redhat 8 we downloaded, select it, say “Open”, then “Ok”.





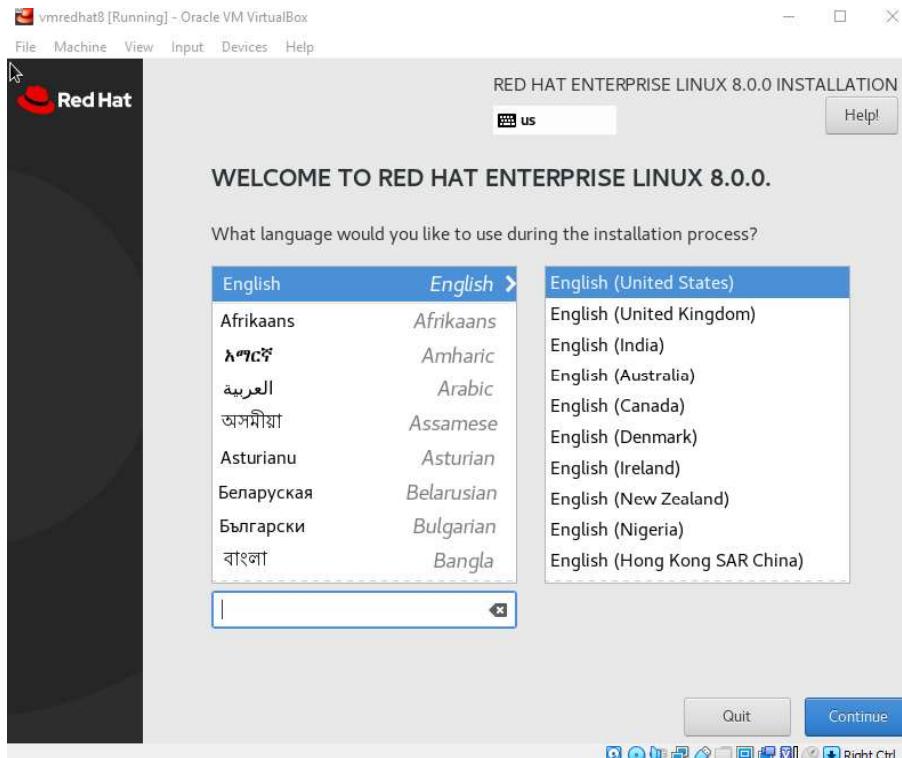
Step 6 – Important step – See that your laptop is connected to internet

Click on vmredhat8, which is the name of my VM and press “Start”... then once click inside the box and using arrows of keyboard go to “Install Red Hat Ent.....”

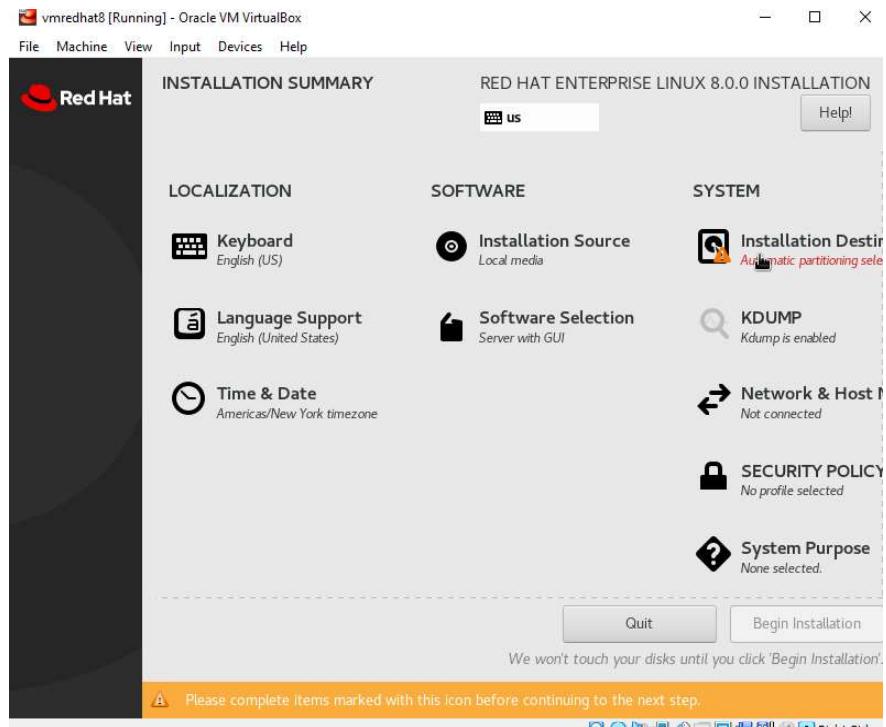


Step 7 – Long step

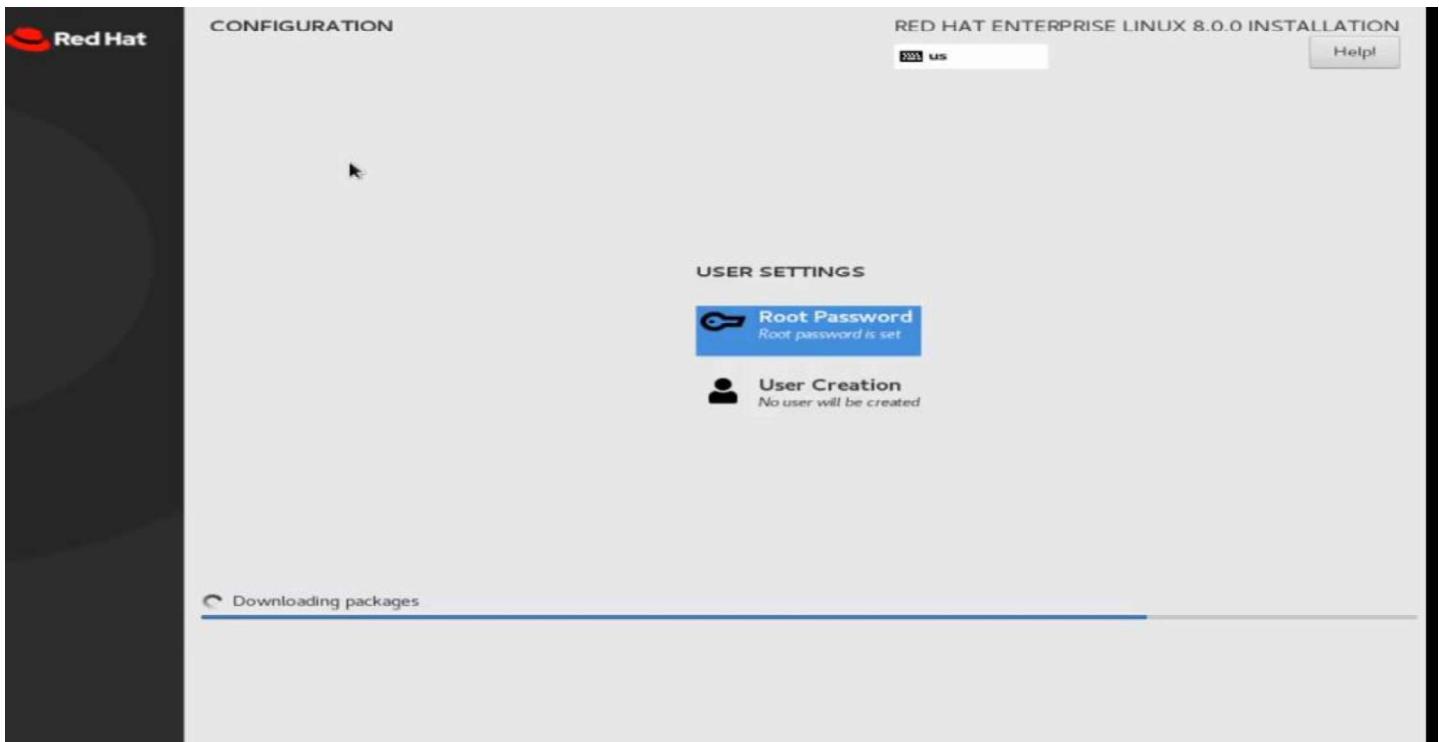
Press “Continue” for English language...



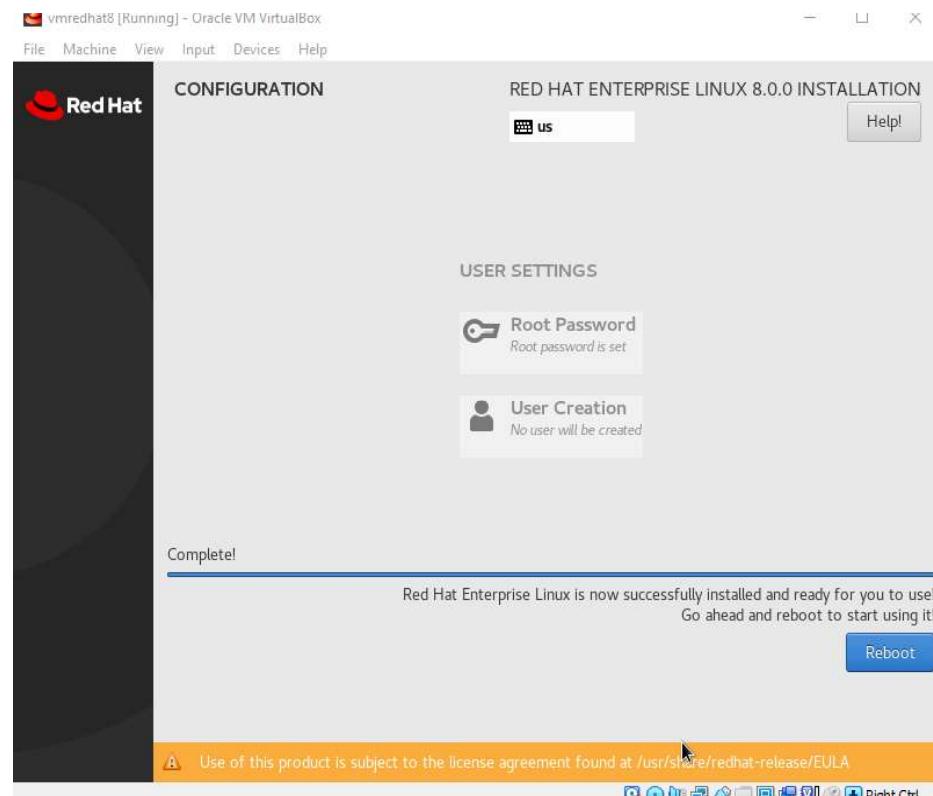
Important - Go to “**Installation Destination**” and press “Done”, then go to “**Network & Host**” and turn Ethernemt to “On”->“Done”, then go to “**Software Selection**” and select “Workstation”->“Done”, then go to “**Time and Date**” and press on India in map->Network time:“On”->“Done”, Indian time will be selected, then say “**Begin Installation**”



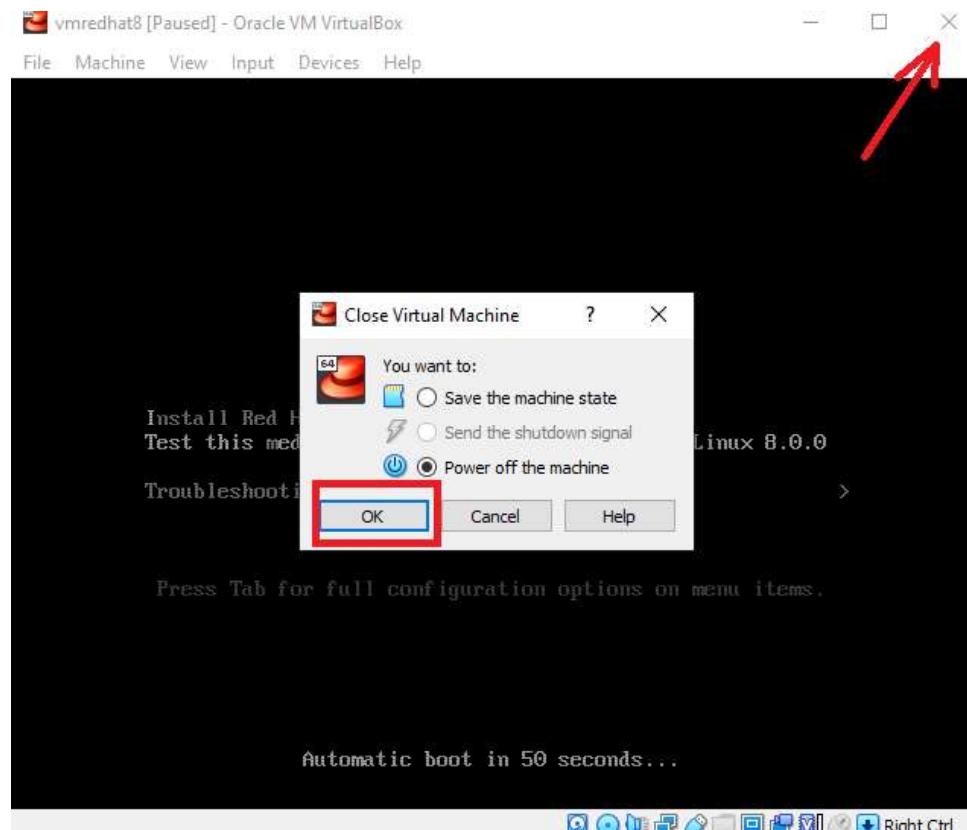
Now click on “Root Password”, set password as “redhat” or anything you like and then **press “Done” twice**, it will take some half hour for installation....



Once the installation is done the following will appear... **Note : After clicking on “Reboot” you have to hard shut down by closing redhat window**

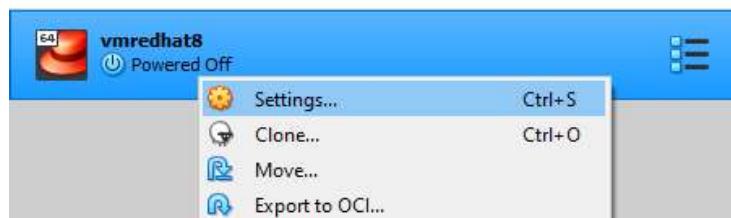


Note : After clicking on “Reboot” you have to **hard shutdown by closing redhat window**



Step 8 –

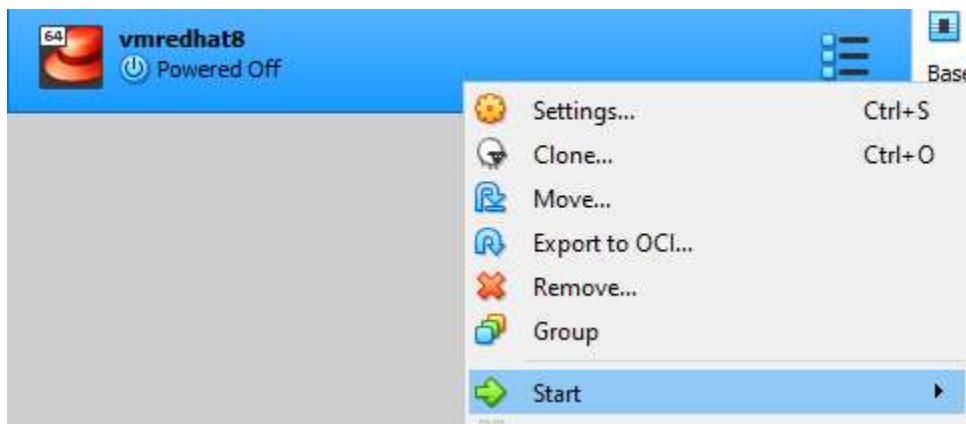
After power off, right click on Redhat OS and go to “Settings...”



Go to “System” and now change the boot order... for that select “HardDisk” and move it to top as follows... then press “OK”

System BEFORE ->		System LATER ->	
General		General	
System		System	
Display		Display	
Storage		Storage	
Audio		Audio	
Network		Network	
Serial Ports		Serial Ports	
USB		USB	
Motherboard		Motherboard	
Processor		Processor	
Acceleration		Acceleration	
Base Memory: 4 MB		Base Memory: 4 MB TOP	
Boot Order:		Boot Order:	
<input checked="" type="checkbox"/> Floppy <input checked="" type="checkbox"/> Optical <input checked="" type="checkbox"/> Hard Disk <input type="checkbox"/> Network		<input checked="" type="checkbox"/> Hard Disk <input checked="" type="checkbox"/> Floppy <input checked="" type="checkbox"/> Optical <input type="checkbox"/> Network	
Move Up		Move Up	
Chipset: PIIX3		Chipset: PIIX3	
Pointing Device: PS/2 Mouse		Pointing Device: PS/2 Mouse	

Now complete installation of RedHat 8 is done... Right click on OS name and press "Start" -> "Normal Start" ... Follow this every time..



Note important :

Click on "License Information" -> check check-box ->"Done" -> "Finish Configuration"... then it will ask to some things.. do next-next or press "skip" .. then they will ask you to create account.. create your name account and Start

Now whenever the system will start you will see login screen... press on **Not listed? ->(username) root -> (password)redhat** (as password we used before while configurations)

Press Ctrl (right)to get windows mouse control... Ctrl(right)+F for full screen and vise versa...

* Take mouse to top left corner.. click on "**Activities**"..go to :: - show applications -> settings -> Devices and Display -> **Resolution -> Select last fourth**..1280x768 and if the access button dosent appear press tab... Tab->Cancel -> one more tab - >press enter -> Keep changes.... You need to do this every time

Done

2

2. Now In this we will configure “**yum**” in fresh installed Red Hat 8

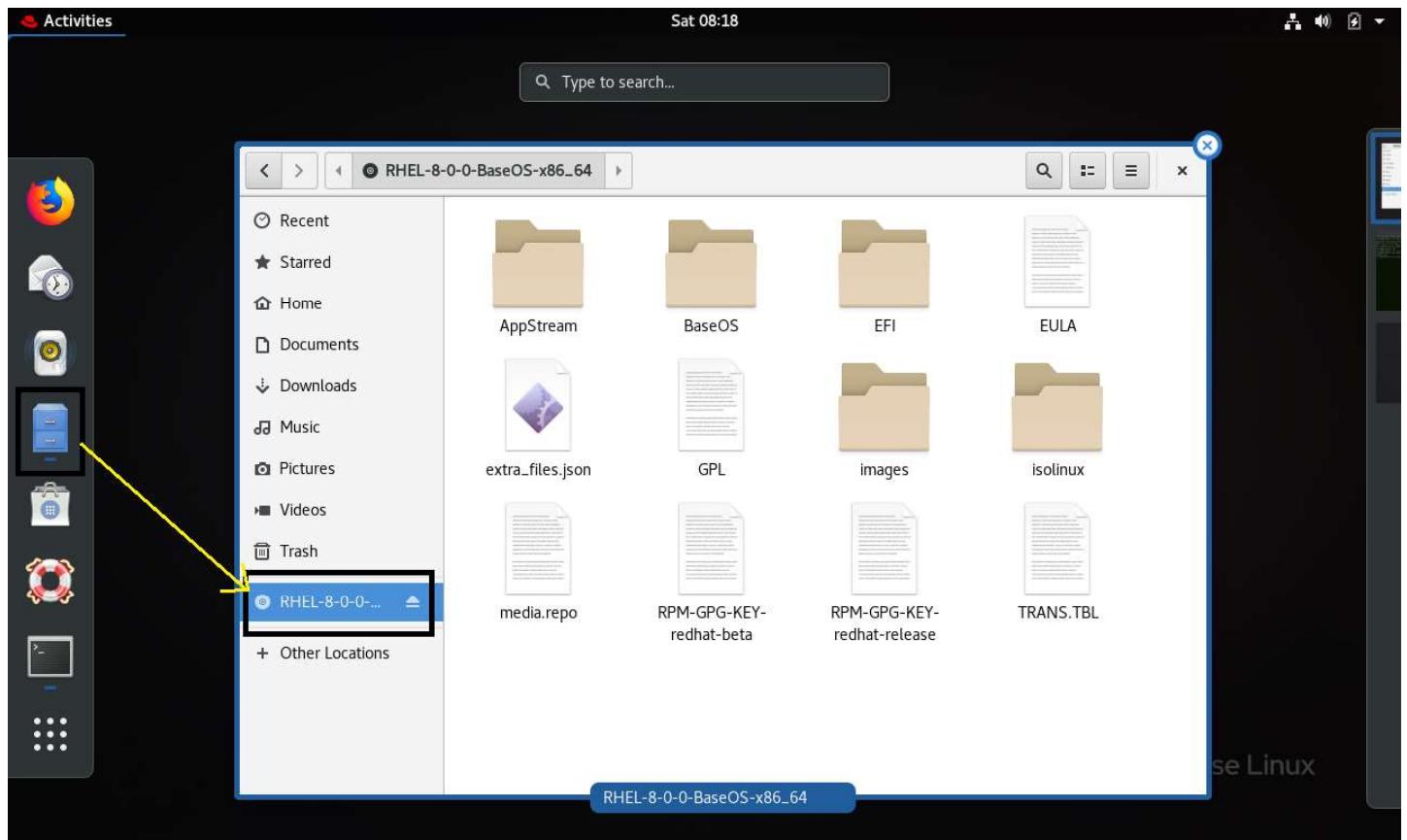
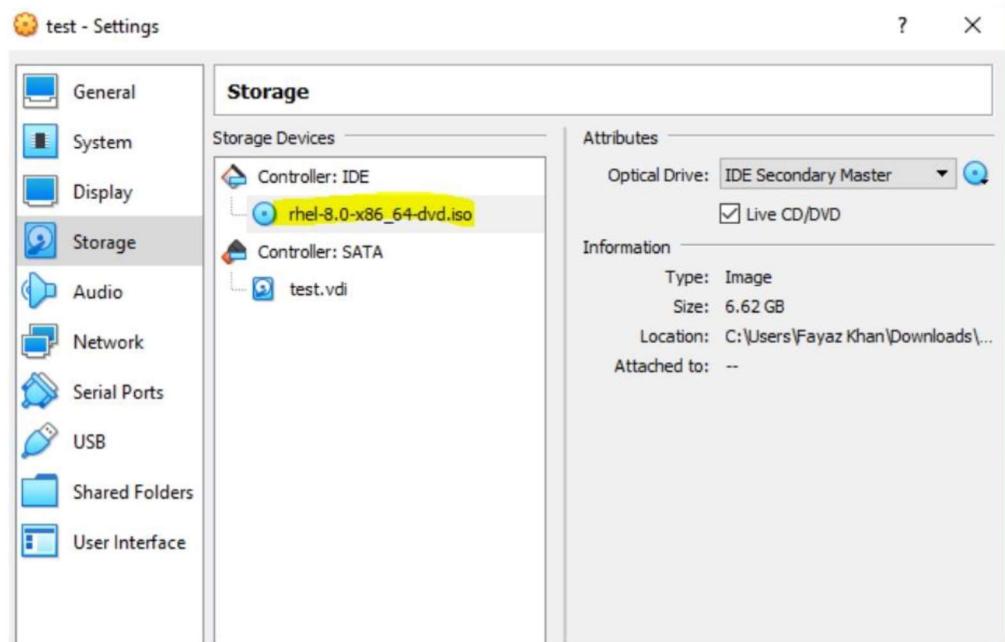
Yum helps to make installation of software very easy as it automatically installs the compatible software from all provided possible locations(links or local storages)

Step 1 –

You should have Red Hat .iso file in “Files” of Red Hat OS. After checking close it...

If don't have do following... It is Oracle VM.. if not there then browse it from windows.. **Add optical drive.. 1st disk**

Attached your Redhat ISO image in your VM :



Step 2 –

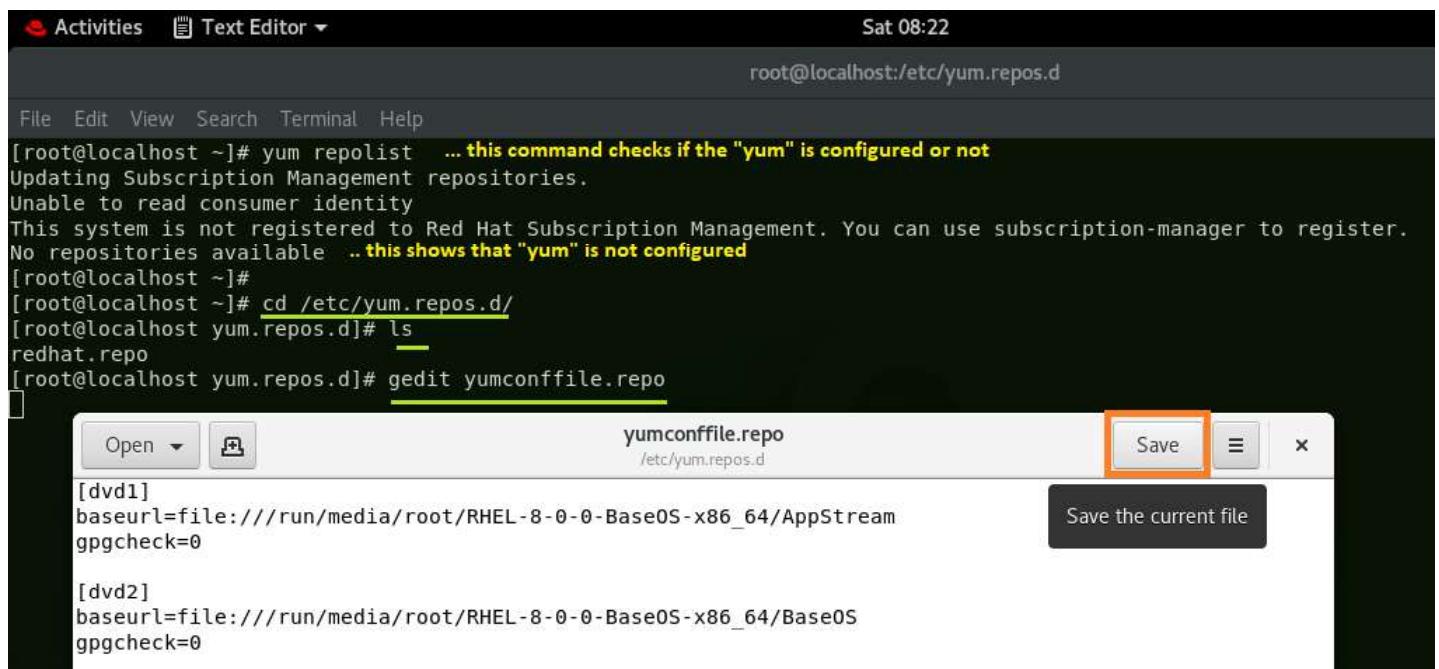
Open terminal and type the following –

Note: while login we have logged in with root so by default root terminal comes up

```
yum repolist  
cd /etc/yum.repos.d/  
ls  
gedit yumconffile.repo
```

Now the text editor opens up where you need to type the following -> save -> close the text editor –

```
[dvd1]  
baseurl=file:///run/media/root/RHEL-8-0-0-BaseOS-x86_64/AppStream  
gpgcheck=0  
  
[dvd2]  
baseurl=file:///run/media/root/RHEL-8-0-0-BaseOS-x86_64/BaseOS  
gpgcheck=0
```



Then again check if yum is installed or not...

```
ls /  
yum repolist
```

Activities Terminal Sat 08:23 root@localhost:/

```
File Edit View Search Terminal Help
[root@localhost ~]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
No repositories available
[root@localhost ~]#
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# ls
redhat.repo
[root@localhost yum.repos.d]# gedit yumconffile.repo
[root@localhost yum.repos.d]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
dvd1                                32 MB/s | 5.3 MB   00:00
dvd2                                19 MB/s | 2.2 MB   00:00
Last metadata expiration check: 0:00:02 ago on Sat 02 May 2020 08:23:05 AM IST.
repo id                           repo name          status
dvd1                               4,672
dvd2                               1,658
[root@localhost yum.repos.d]# cd / .. it takes you to the absolute folder of Red Hat
[root@localhost /]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
Last metadata expiration check: 0:00:23 ago on Sat 02 May 2020 08:23:05 AM IST.
repo id                           repo name          status
dvd1                               4,672
dvd2                               1,658
[root@localhost /]#
```



The above shows that we have successfully configured “yum”

Done

3

3. Installing Docker using yum

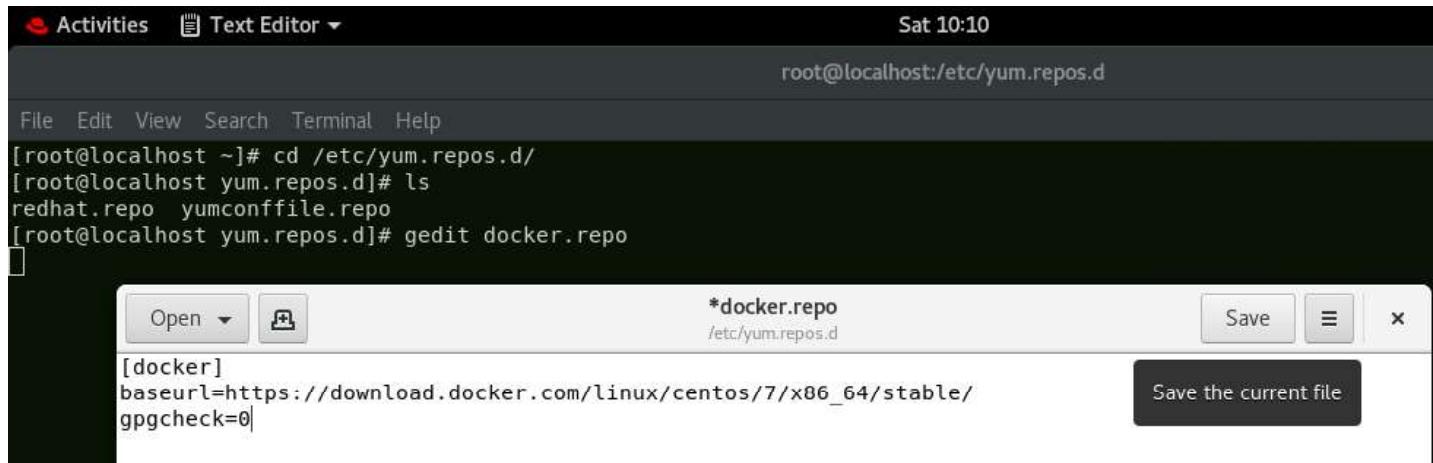
Step 1 –

Open Terminal and type the following, save and close –

```
cd /etc/yum.repos.d/  
ls  
gedit docker.repo
```

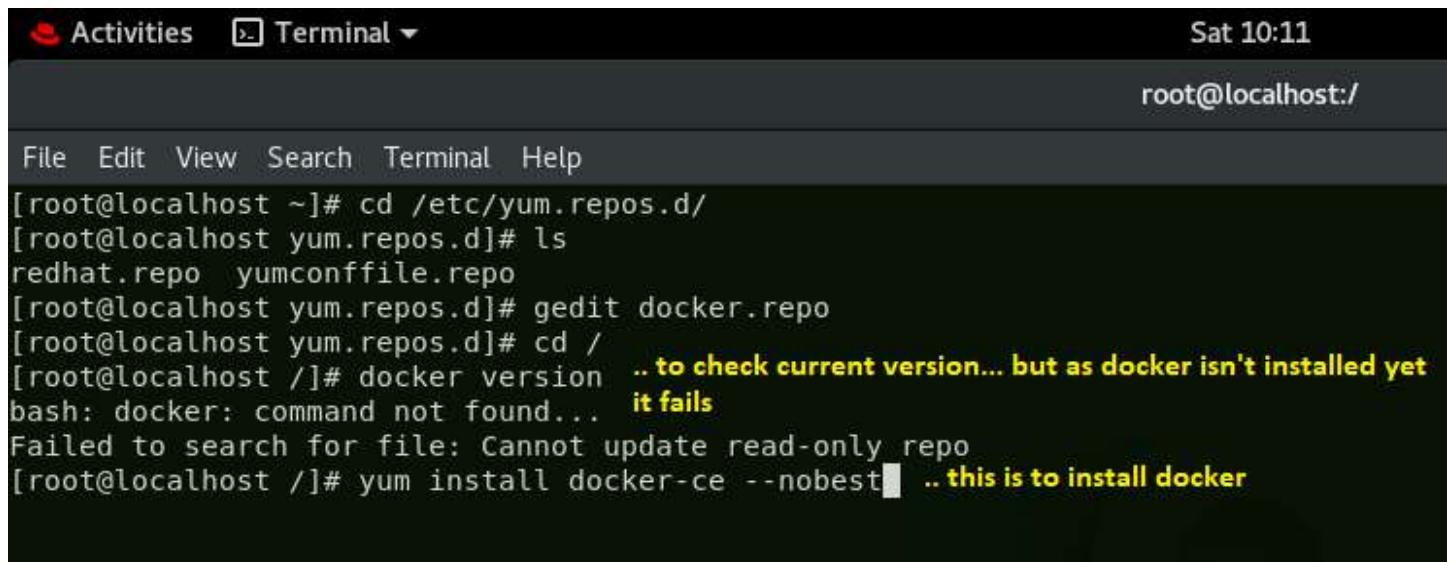
Now write the following in the opened file –

```
[docker]  
baseurl= https://download.docker.com/linux/centos/7/x86_64/stable/  
gpgcheck=0
```



Step 2 - Now save and close...then type the following –

```
cd /  
docker version  
yum install docker-ce --nobest
```



It will take 10-15 minutes to install... in between it will ask to type "y"

```
Total size: 80 M
Total download size: 80 M
Installed size: 339 M
Is this ok [y/N]: y
```

Now after installation you can check if docker is installed or not... using command - `docker version`

```
docker: error while loading shared libraries: libltdl.so.0: cannot open shared object file: No such file or directory
Complete!
[root@localhost /]# docker version
Client: Docker Engine - Community
  Version:           19.03.8
  API version:        1.40
  Go version:         go1.12.17
  Git commit:        afacb8b
  Built:              Wed Mar 11 01:27:
  OS/Arch:            linux/amd64
  Experimental:      false
```

Done



4. Installing docker compose

Docker Compose is used to run multiple containers as a single service. For example, suppose you had an application which required NGNIX and MySQL, you could create one file which would start both the containers as a service without the need to start each one separately.

Here there is a single step –

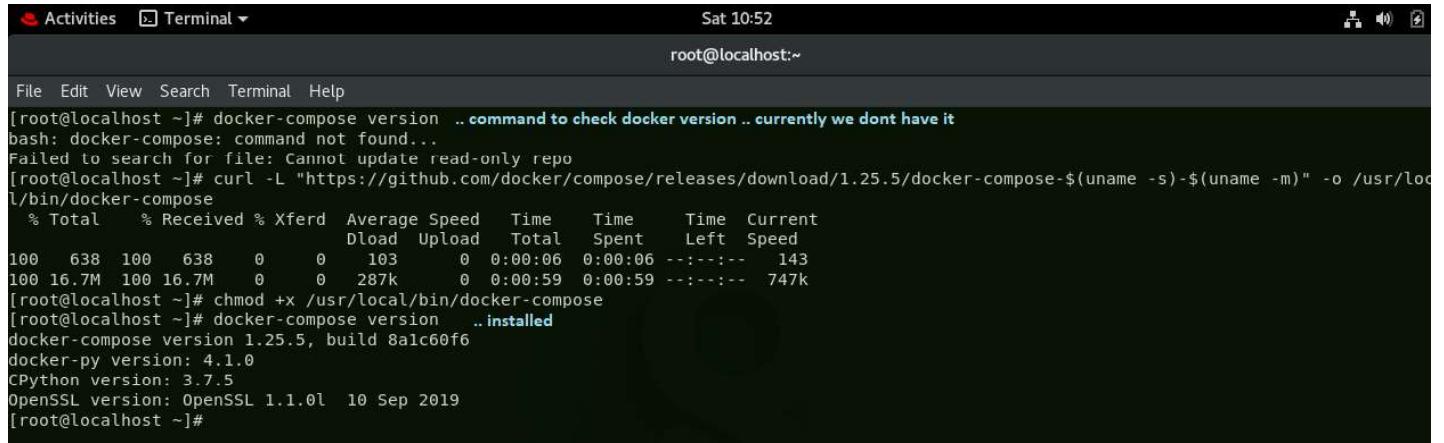
Fire the following commands on Terminal

docker-compose verison

```
curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

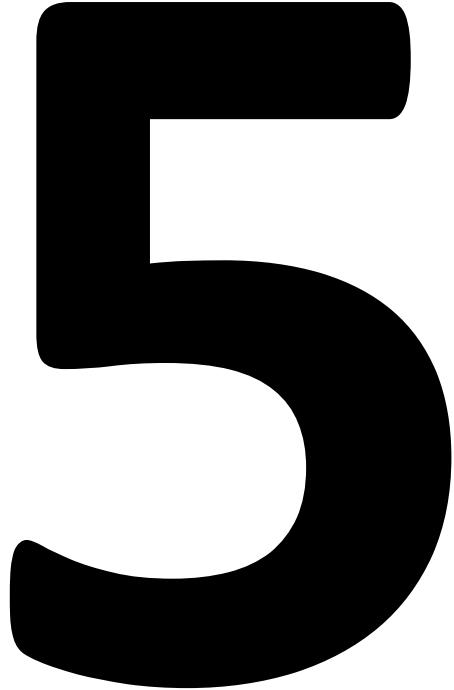
docker-compose verison



The screenshot shows a terminal window titled 'Activities' with a 'Terminal' tab selected. The window title bar includes icons for battery, signal, and volume. The status bar at the top right shows 'Sat 10:52'. The terminal prompt is 'root@localhost:~'. The terminal window contains the following text:

```
File Edit View Search Terminal Help
[root@localhost ~]# docker-compose version ..command to check docker version ..currently we dont have it
bash: docker-compose: command not found...
Failed to search for file: Cannot update read-only repo
[root@localhost ~]# curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent    Left  Speed
100  638  100  638    0     0  103      0  0:00:06  0:00:06  --:--:--  143
100 16.7M  100 16.7M   0     0  287k     0  0:00:59  0:00:59  --:--:--  747k
[root@localhost ~]# chmod +x /usr/local/bin/docker-compose
[root@localhost ~]# docker-compose version ..installed
docker-compose version 1.25.5, build 8a1c60f6
docker-py version: 4.1.0
CPython version: 3.7.5
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
[root@localhost ~]#
```

Done



5. Building Infrastructure(for owncloud) with persistent storage so if the website crashes we will be able to launch new server on the top of docker and will link same persistent storage.
It takes less than 1 second to configure a complete new server.

Now you need to get smart, as till now you have became use-to with Red Hat and its terminal... execute following commands as executed

```
Activities Terminal Thu 2:10 PM
root@localhost:~>

File Edit View Search Terminal Help
[root@localhost ~]# docker volume ls ....using this command we can check the persistent (permanant) storages
DRIVER      VOLUME NAME
[root@localhost ~]# docker ps -a .... it gives current operating systems in our docker container ... now there is none
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS        PORTS     NAMES
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS        PORTS     NAMES
[root@localhost ~]# systemctl restart docker
[root@localhost ~]# docker run -dit -v owncloud_persistent_db:/var/www/html/data -p 8080:80 --name ocos owncloud:latest
c2c5b0046ad7a950bd37c17ff445126555a093d0998c142e19d7bd52fe3e74f9
[root@localhost ~]# docker ps ... it gives the current running operating systems in our docker container
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS        PORTS     NAMES
c2c5b0046ad7   owncloud:latest "docker-entrypoint.s..." 16 seconds ago Up 13 seconds
[root@localhost ~]# ifconfig enp0s3 .. it gives IP address of our redhat which is running on virtual machine
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.43.115 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 2405:204:907c:96d9:2d12:4587:7a1b:f9e3 prefixlen 64 scopeid 0x0<global>
inet6 fe80::5b6e:1887:f116:2492 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:4c:6e:8a txqueuelen 1000 (Ethernet)
RX packets 204596 bytes 272222510 (259.6 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 83195 bytes 6563597 (6.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@localhost ~]#
```

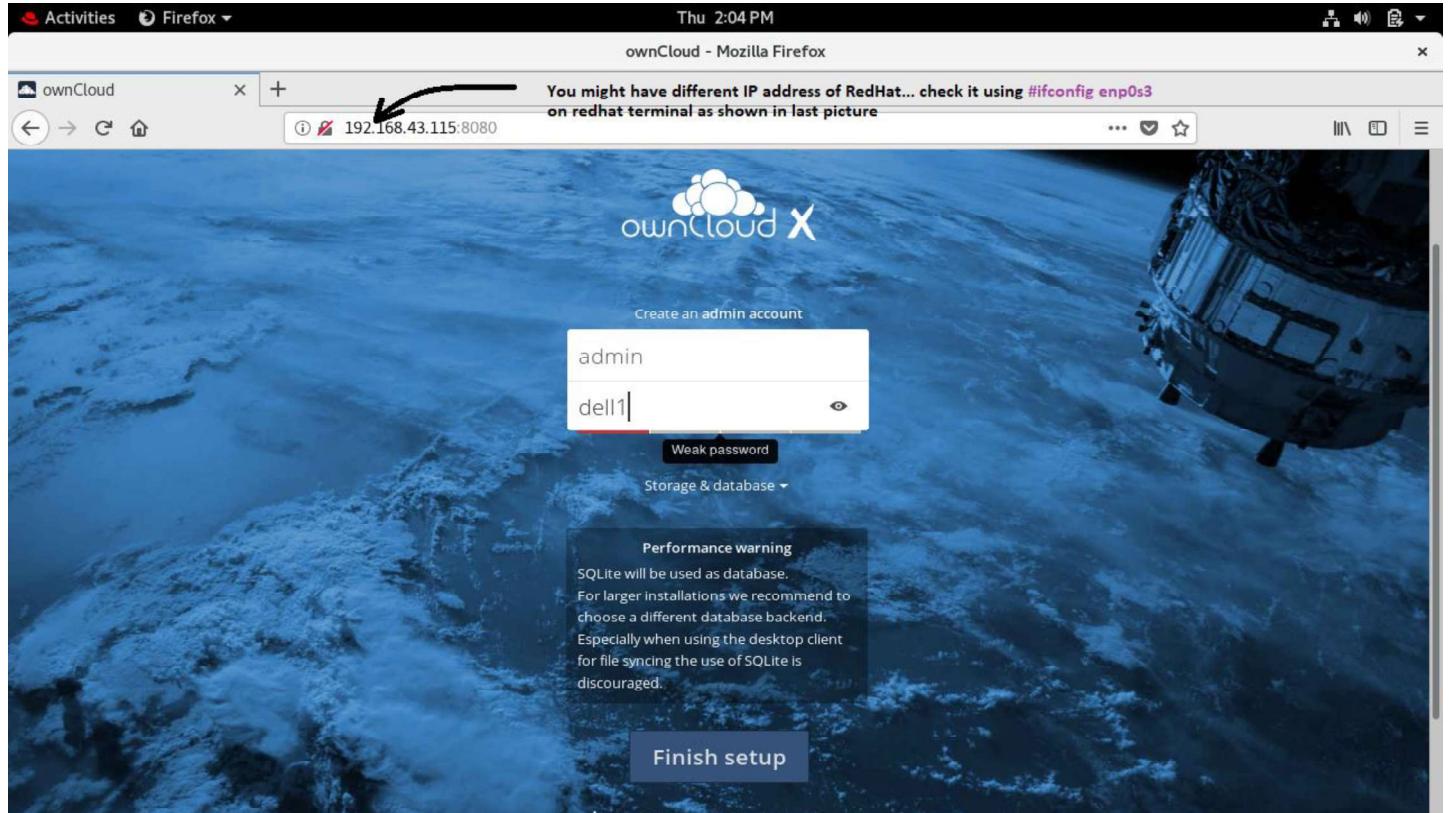
This is the IP address of our server which we will be using to access our owncloud server....

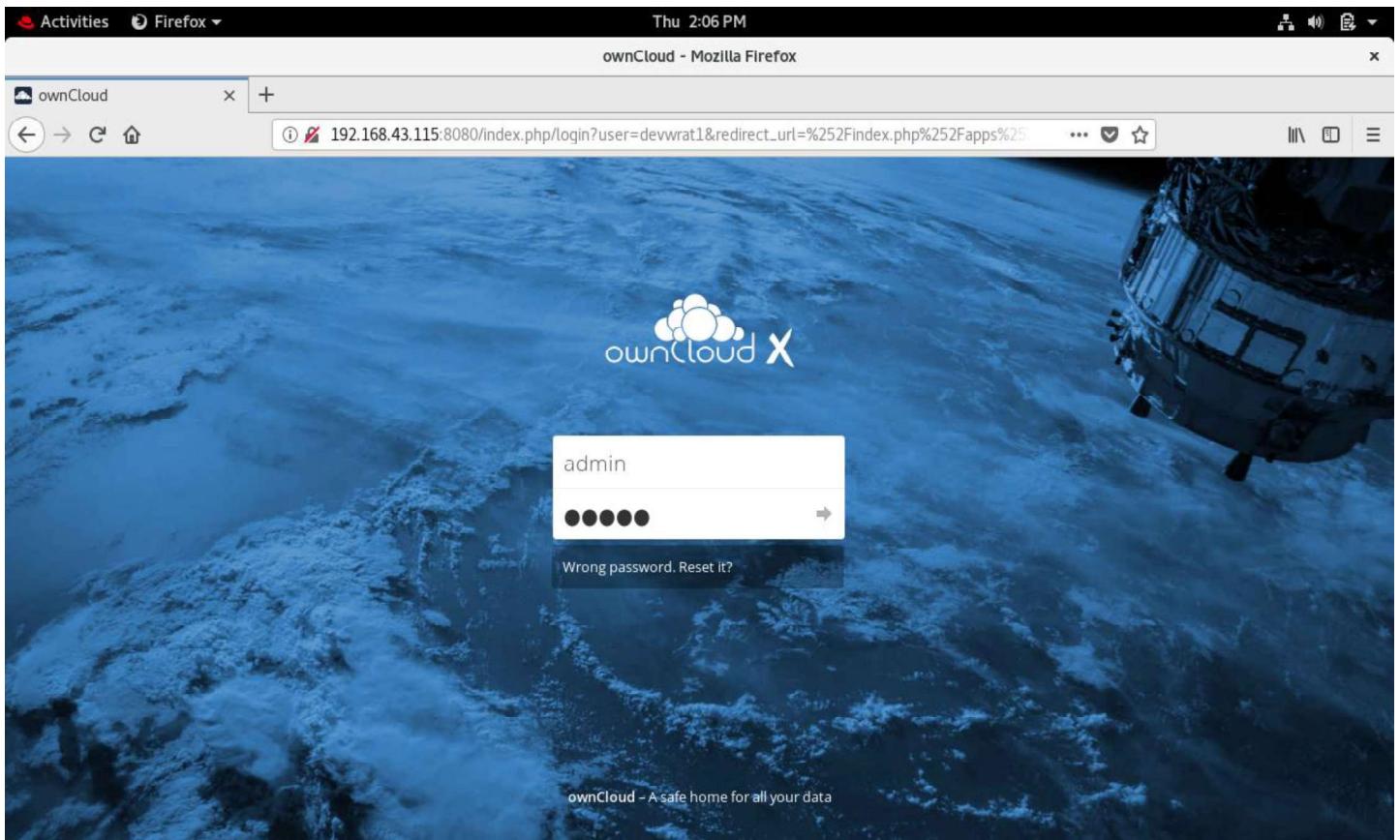
We will use it as 192.168.43.115:8080

...here 8080 is the port number to identify the owncloud services

This will create a new persistent storage in our base system so that if the OS on docker container crashes we won't lose our data...
The command also creates the OS to run on docker... it configures OS and runs it..

Now instead of below IP go to your IP address as – ip:8080





You can play with the UI and can learn to do multiple things...

A screenshot of a Mozilla Firefox browser window showing the ownCloud file manager interface. The URL in the address bar is 192.168.43.115:8080/index.php/apps/files/?dir=/&fileId=3. The left side of the screen has a sidebar with links for 'All files', 'Favorites', 'Shared with you', 'Shared with others', 'Shared by link', and 'Tags'. The right side shows a list of files and folders. At the top, there's a header with the ownCloud logo and a user profile for 'admin'. Below the header, there's a toolbar with icons for home, back, forward, and search. The main area lists two folders, 'Documents' and 'Photos', which are both highlighted with a green rectangular selection box. Other items listed include 'ownCloud Manual.pdf' and some small icons. At the bottom of the main area, it says '2 folders and 1 file' and '5.4 MB'. At the very bottom of the screen, there are links for 'Deleted files' and 'Settings'.

Now I am going to photos folder and uploading the start screen screenshot.. later I will delete container and see if the screenshot is obtained back on another container web-server

Thu 2:07 PM

Photos - Files - ownCloud - Mozilla Firefox

192.168.43.115:8080/index.php/apps/files/?dir=/Photos&fileId=6

All files

Paris.jpg

San Francisco.jpg

Squirrel.jpg

3 files

Deleted files

192.168.43.115:8080/index.php/apps/files/?dir=/Photos&fileId=6#

Thu 2:08 PM

Photos - Files - ownCloud - Mozilla Firefox

192.168.43.115:8080/Index.php/apps/files/?dir=/Photos&fileId=6

All files

Paris.jpg

San Francisco.jpg

Screenshot from 2020-04-30 14-04-12.png

Squirrel.jpg

4 files

Deleted files

Settings
192.168.43.115:8080/remote.php/webdav/photos/Screenshot from 2020-04-30 14-04-12.png

You can also visit to users and add users, groups, etc

Screenshot of the ownCloud Files interface in Mozilla Firefox. The URL is 192.168.43.115:8080/index.php/apps/files/?dir=/&fileid=3. The sidebar shows navigation links like All files, Favorites, Shared with you, Shared with others, Shared by link, Tags, and Deleted files. The main area lists three items: 'Documents' (35 KB), 'Photos' (1.7 MB), and 'ownCloud Manual.pdf' (4.7 MB, 3 minutes ago). A sidebar on the right shows 'Settings' with 'Users' selected, 'Help', and 'Log out'. The status bar at the bottom shows the URL 192.168.43.115:8080/index.php/settings/users.

Screenshot of the ownCloud Users interface in Mozilla Firefox. The URL is 192.168.43.115:8080/index.php/settings/users#. The sidebar shows 'Users' and '+ Add Group'. The main area shows a table of users. The first user listed is 'Everyone' (admin) with the following details: Username: admin, E-Mail: (redacted), Groups: admin, Group Admin for: no group, and Quota: Default. The 'Groups' and 'Create' buttons are highlighted with blue boxes. The status bar at the bottom shows the URL 192.168.43.115:8080/index.php/settings/users#.

	Username	E-Mail	Groups	Create	Group Admin for	Quota
Everyone	admin	(redacted)	admin		no group	Default
Admins						

You can go back from files... You can just play with UI and learn...

The screenshot shows a Firefox browser window titled "Users - ownCloud - Mozilla Firefox". The URL in the address bar is "192.168.43.115:8080/index.php/settings/users". The page displays a list of users under the heading "Users". There is one user entry for "admin" with the following details:

Username	Full Name	Password	Groups	Group Admin for	Quota
admin	admin	*****	admin	no group	Default

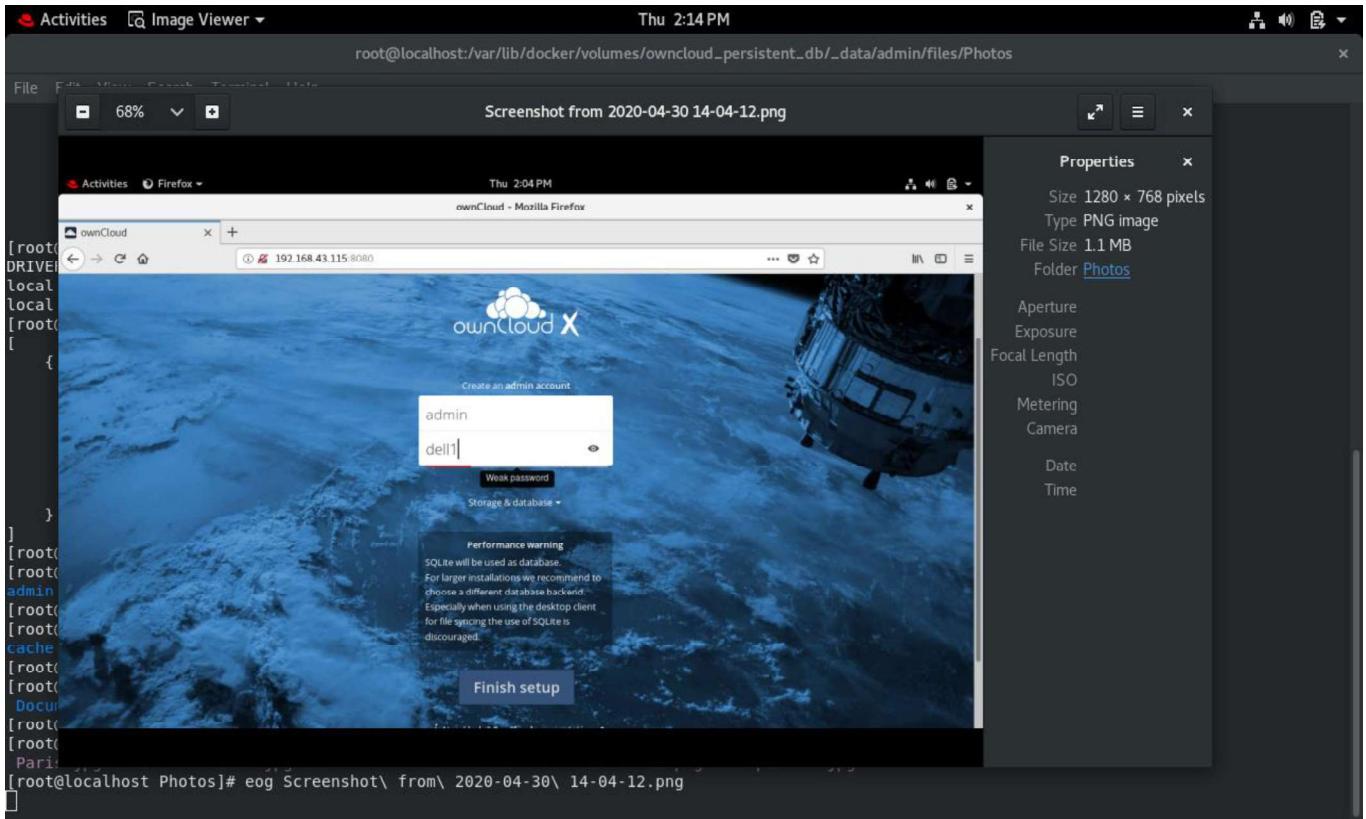
The "Files" icon in the sidebar is highlighted with a blue box.

192.168.43.115:8080/index.php/settings/users#

On terminal we can enter following commands and view the screenshot uploaded on persistent (permanent) storage of server

```
root@localhost:~# docker volume ls
DRIVER    VOLUME NAME
local     fa3fae35b3113fe5263768ba7a5747611bc40da2e72d83171efb770b4ce1f601
local     owncloud_persistent_db
[root@localhost ~]# docker volume inspect owncloud_persistent_db
[
  {
    "CreatedAt": "2020-04-30T14:11:44+05:30",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/owncloud_persistent_db/_data",
    "Name": "owncloud_persistent_db",
    "Options": null,
    "Scope": "local"
  }
]
[root@localhost ~]# cd /var/lib/docker/volumes/owncloud_persistent_db/_data
[root@localhost _data]# ls
admin_avatars index.html owncloud.db owncloud.log
[root@localhost _data]# cd admin
[root@localhost admin]# ls
cache files thumbnails
[root@localhost admin]# cd files
[root@localhost files]# ls
Documents 'ownCloud Manual.pdf' Photos
[root@localhost files]# cd Photos/
[root@localhost Photos]# ls
Paris.jpg 'San Francisco.jpg' 'Screenshot from 2020-04-30 14-04-12.png' Squirrel.jpg
[root@localhost Photos]# eog Screenshot from 2020-04-30 14-04-12.png
```

This is the following screenshot stored...



Now we will delete the running server which is like the server is crashing and services are stopped....

You can check the browser and refresh page... you will find that server is down n....

Now will again run new server using the persistent storages... we will login from browser once the server is configure, and will see the browser with same account where we uploaded browser

The screenshot shows a terminal window with the title 'Activities Terminal'. The terminal session is as follows:

```
root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
c2c5b0046ad7        owncloud:latest     "docker-entrypoint.s..."   24 minutes ago    Exited (0) 55 seconds ago          ocos

[root@localhost Photos]# docker rm -f ocos
ocos

[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc55826088f33134ecd7d845ba958727bee97bba6c966a35288d84c868
[root@localhost Photos]# docker run -dit -v owncloud_persistent_db:/var/www/html/data -p 8080:80 --name ocos owncloud:latest
OS name installed on top of docker
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest     "docker-entrypoint.s..."   39 seconds ago    Up 36 seconds      0.0.0.0:8080->80/tcp   ocos

[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest     "docker-entrypoint.s..."   5 minutes ago     Up 5 minutes       0.0.0.0:8080->80/tcp   ocos

[root@localhost Photos]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest     "docker-entrypoint.s..."   5 minutes ago     Up 5 minutes       0.0.0.0:8080->80/tcp   ocos

[root@localhost Photos]# docker stop ocos ... to stop the running os Now the server will stop working if you check ocos
...to remove OS from docker container... in real world it is same as some server crashes and the service stops
```

A yellow annotation with an arrow points to the text 'OS name installed on top of docker' in the terminal output. Another annotation with a blue box highlights the word 'ocos' in the 'NAMES' column of the final 'docker ps' command.

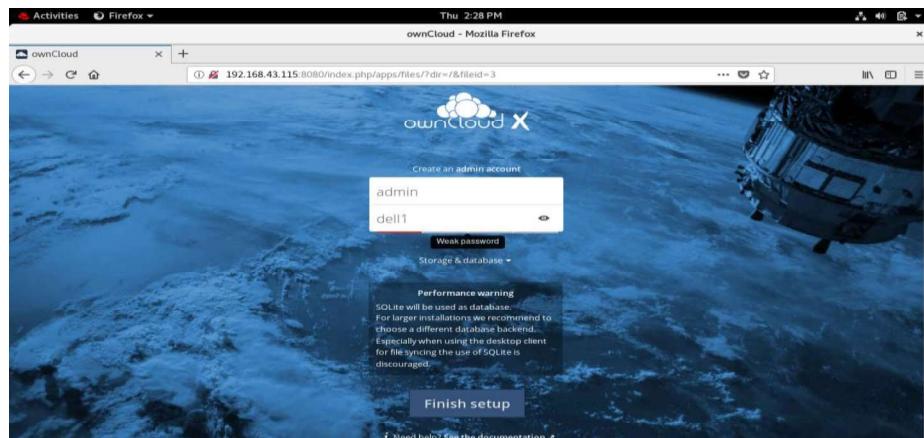
The screenshot shows a Linux desktop environment with a window titled "Files - ownCloud - Mozilla Firefox". The browser is displaying a list of files in a folder named "Photos". A yellow box highlights a message at the top of the page: "Failed to request notifications. Please try to refresh the page manually. Problem loading page, reloading in 5 seconds". The terminal window below shows Docker commands being run:

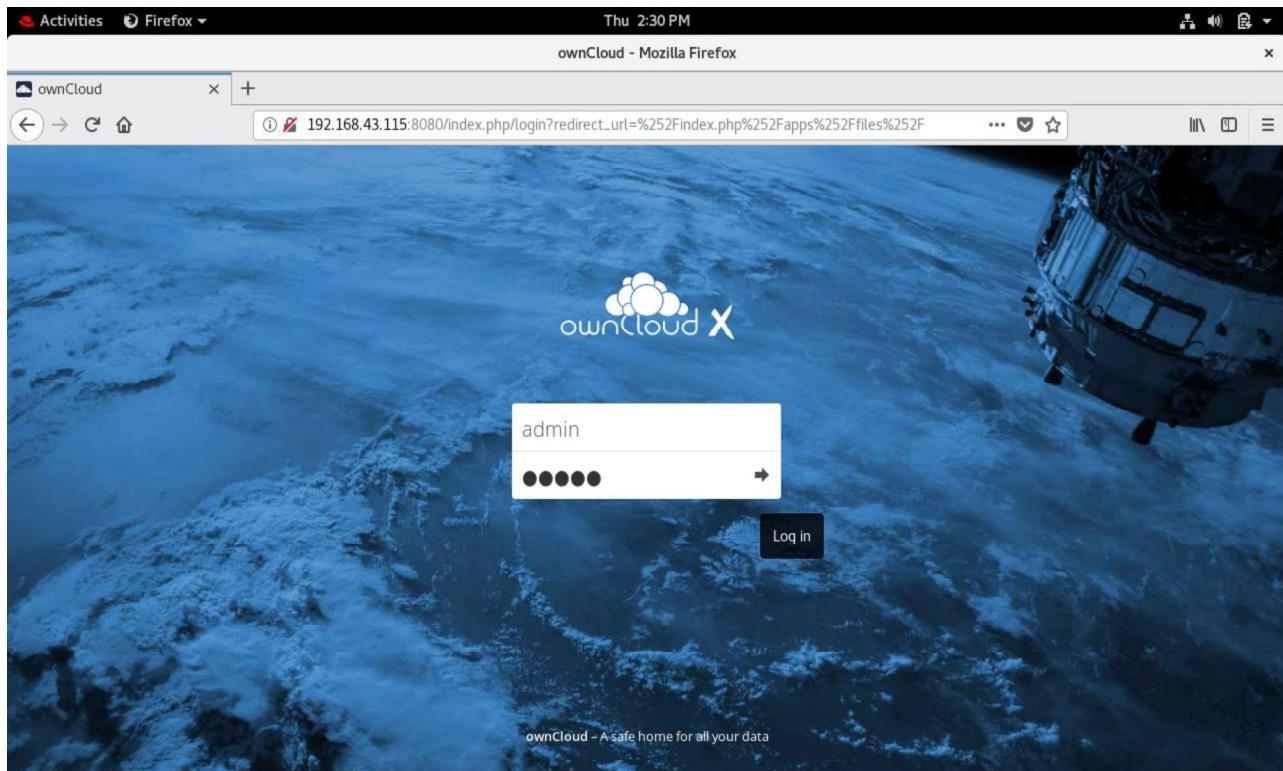
```

[ec58001000]  owncloud:latest    "/usr/bin/docker-entrypoint.sh"   24 minutes ago   EXITED (0, 33 seconds ago)   0.0.0.0:8080
[root@localhost Photos]# docker rm -f ocos
ocos
[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
16f9d6cc55826088f3134ecd7d845ba958727bee97bb9a6c966a3528d84c868
[root@localhost Photos]# docker run -dit -v owncloud_persistent_db:/var/www/html/data -p 8080:80 --name ocos owncloud:latest
[root@localhost Photos]# docker ps ... we again created and launched our new os on top of container... again enter same username and password, services will be there
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
16f9d6cc5582       owncloud:latest     "docker-entrypoint.s..."   39 seconds ago   Up 36 seconds   0.0.0.0:8080->80/tcp   ocos
[root@localhost Photos]#

```

Logging in again.... Using same id - password



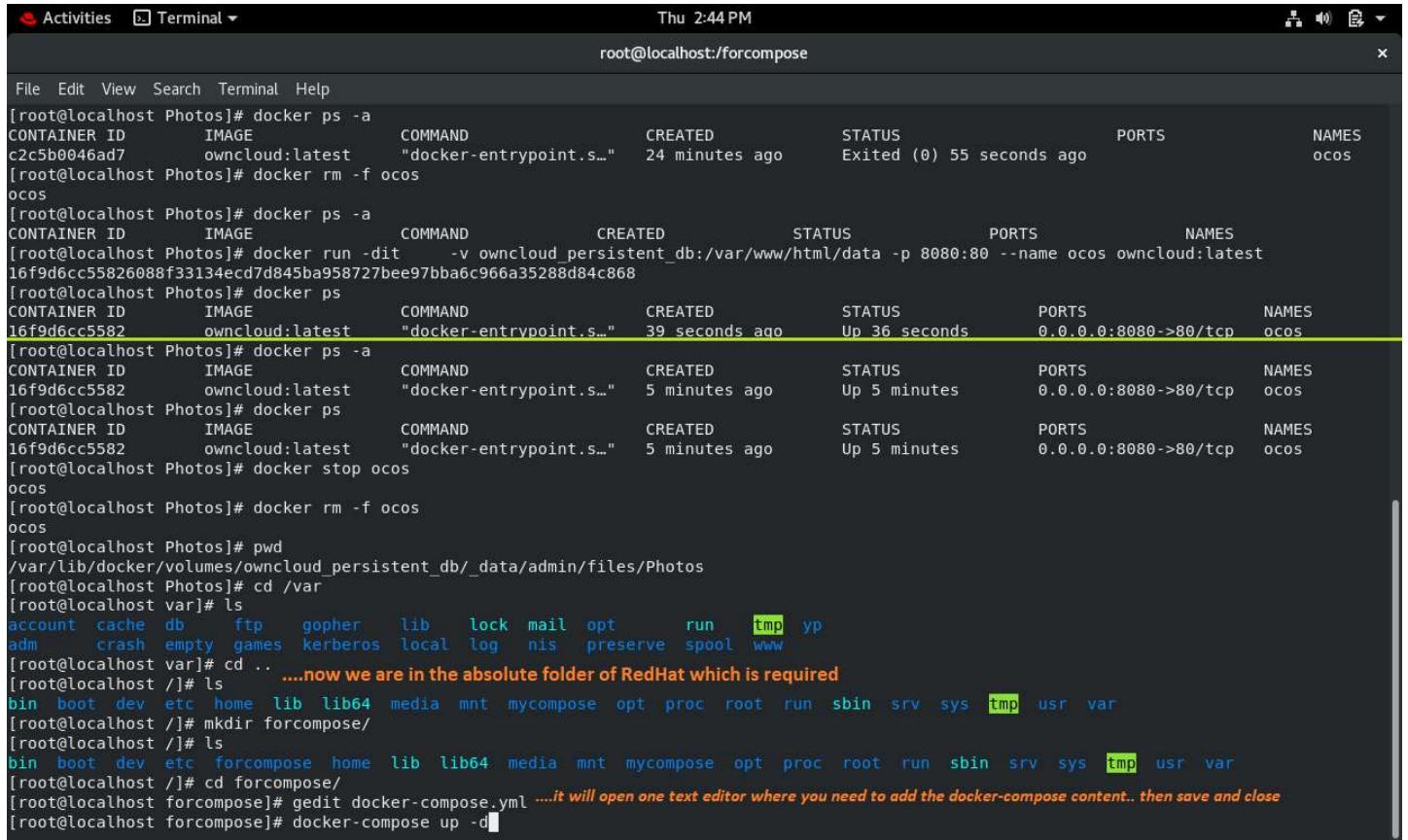


And here you finds your image on new server too... this is how it works in real life...

A screenshot of a Firefox browser window showing the ownCloud file list. The title bar indicates "Photos - Files - ownCloud - Mozilla Firefox" and the time "Thu 2:30 PM". The left sidebar shows navigation options like "All files", "Favorites", "Shared with you", "Shared with others", "Shared by link", and "Tags". The main area displays a list of four files: "Paris.jpg" (223 KB, seconds ago), "San Francisco.jpg" (211 KB, seconds ago), "Screenshot from 2020-04-30 14-04-12.png" (1.1 MB, 27 minutes ago), and "Squirrel.jpg" (228 KB, seconds ago). The total size of the files is 1.7 MB.

Now we will do this same using docker compose... we have discussed about docker compose on the top of this ppt..

Docker compose requires new folder and docker-compose.yml file, and the following command in the same path –
`docker-compose up -d`



The screenshot shows a terminal window titled "Activities Terminal" with the command "root@localhost:/forcompose". The terminal displays the following sequence of commands and their outputs:

```
[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
c2c5b0046ad7        owncloud:latest    "docker-entrypoint.s..."   24 minutes ago    Exited (0) 55 seconds ago   ocos

[root@localhost Photos]# docker rm -f ocos
ocos

[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
[root@localhost Photos]# docker run -dit      -v owncloud_persistent_db:/var/www/html/data -p 8080:80 --name ocos owncloud:latest
16f9d6cc55826088f33134ecd7d845ba958727bee97bba6c966a35288d84c868
[root@localhost Photos]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest    "docker-entrypoint.s..."   39 seconds ago   Up 36 seconds     0.0.0.0:8080->80/tcp   ocos
[root@localhost Photos]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest    "docker-entrypoint.s..."   5 minutes ago    Up 5 minutes     0.0.0.0:8080->80/tcp   ocos
[root@localhost Photos]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
16f9d6cc5582        owncloud:latest    "docker-entrypoint.s..."   5 minutes ago    Up 5 minutes     0.0.0.0:8080->80/tcp   ocos
[root@localhost Photos]# docker stop ocos
ocos
[root@localhost Photos]# docker rm -f ocos
ocs
[root@localhost Photos]# pwd
/var/lib/docker/volumes/owncloud_persistent_db/_data/admin/files/Photos
[root@localhost Photos]# cd /var
[root@localhost var]# ls
account cache db ftp gopher lib lock mail opt run tmp yp
adm crash empty games kerberos local log nis preserve spool www
[root@localhost var]# cd ..
....now we are in the absolute folder of RedHat which is required
[root@localhost /]# ls
bin boot dev etc home lib lib64 media mnt mycompose opt proc root run sbin srv sys tmp usr var
[root@localhost /]# mkdir forcompose/
[root@localhost /]# ls
bin boot dev etc forcompose home lib lib64 media mnt mycompose opt proc root run sbin srv sys tmp usr var
[root@localhost /]# cd forcompose/
[root@localhost forcompose]# gedit docker-compose.yml ....it will open one text editor where you need to add the docker-compose content.. then save and close
[root@localhost forcompose]# docker-compose up -d
```

Following is the docker compose program which configures everything for you in one click ... **the indentation are spaces not tabs.. this is important to note, same name that is docker-compose.yml is must**

```
version: '2'
services:
  owncloud:
    images: owncloud:latest
    ports:
      - '8080:80'
    volumes:
      - owncloud_persistent_db:/var/www/html/data
volumes:
  owncloud_persistent_db:
```

Note –

`docker-compose stop` -> To stop your application

`docker-compose start` -> To restart application

`docker-compose down` -> To stop and remove containers, networks and images created by the docker-compose.yml

`docker-compose down -volumes` -> This will permanently delete the customizations you've made so far

Activities Text Editor ▾

Thu 2:43 PM

Open docker-compose.yml
/forcompose

```
version: '2'
services:
  owncloud:
    image: owncloud:latest
    ports:
      - '8080:80'
    volumes:
      - owncloud_persistent_db:/var/www/html/data

volumes:
  owncloud_persistent_db:
```

Following commands are to stop the docker compose, then remove it and configure->run back to avail services..

docker-compose stop

docker-compose rm

y

docker-compos up -d

Activities Terminal ▾

Thu 3:11 PM

root@localhost:/forcompose

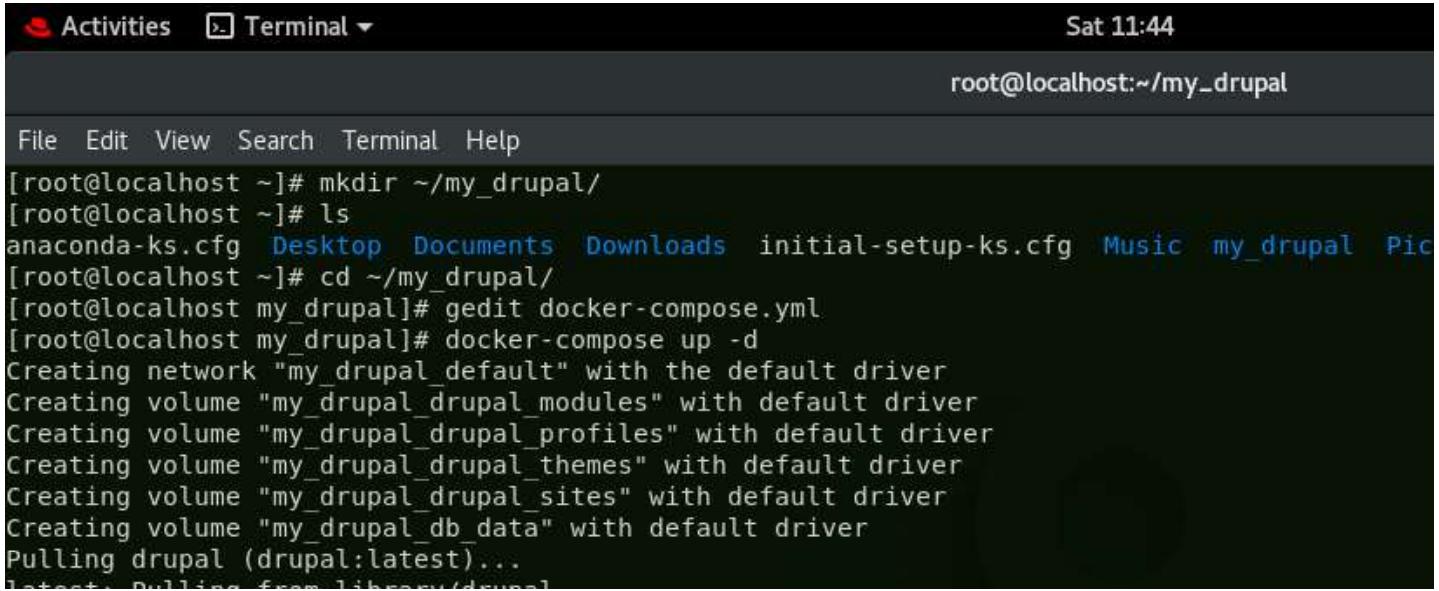
```
File Edit View Search Terminal Help
[root@localhost forcompose]# docker-compose ps
          Name        Command       State     Ports
----- 
forcompose_owncloud_1   docker-entrypoint.sh apache ...   Up      0.0.0.0:8080->80/tcp
[root@localhost forcompose]# docker-compose stop
Stopping forcompose_owncloud_1 ... done
[root@localhost forcompose]# docker-compose rm
Going to remove forcompose_owncloud_1
Are you sure? [yN] y
Removing forcompose_owncloud_1 ... done
.....now the OS contained in the docker container is deleted... consider it as the server is crashed as its a real life use case faced by the organization... now again in 1 click new OS will be created and server will start working again
[root@localhost forcompose]# docker-compose up -d
Creating forcompose_owncloud_1 ... done
[root@localhost forcompose]#
```

Done

6

6. Infrastructure for Drupal

Below you will find some commands in the screenshots just follow them...



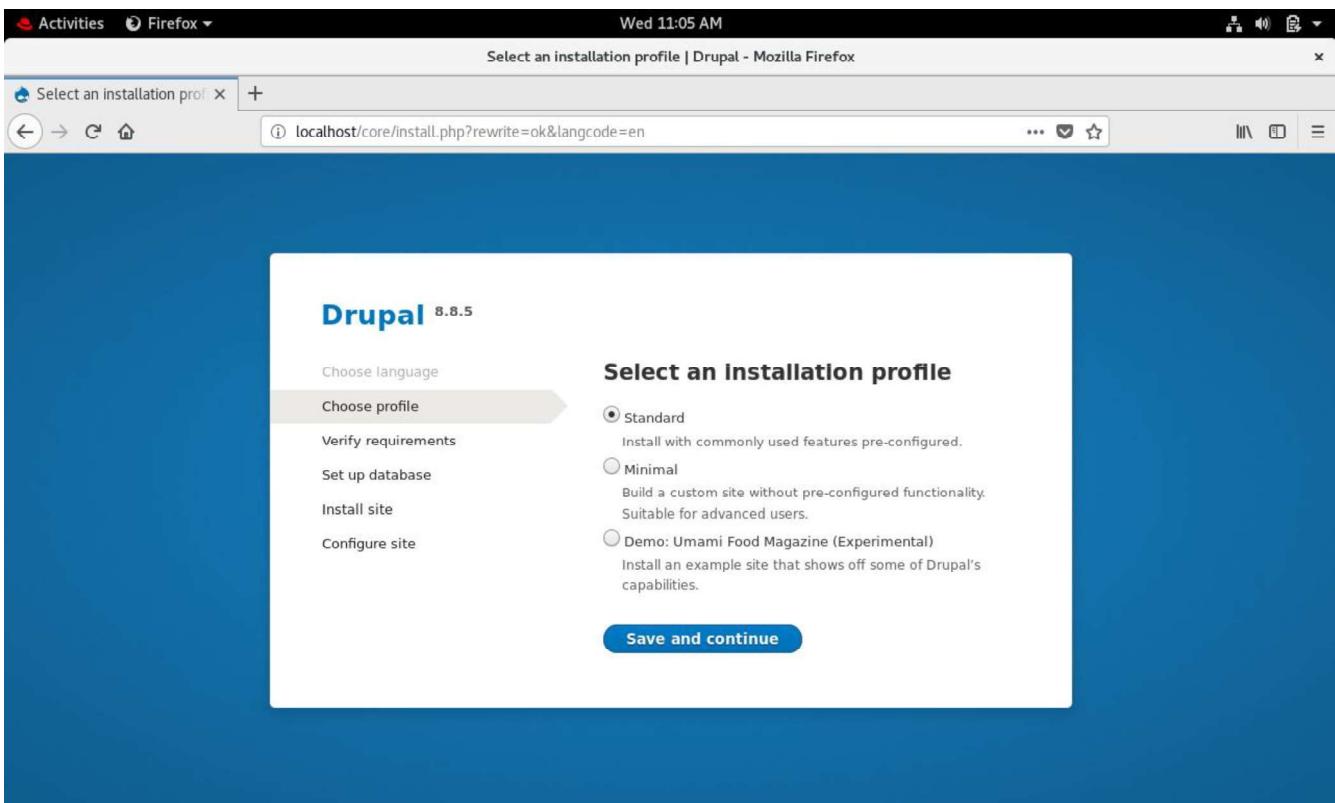
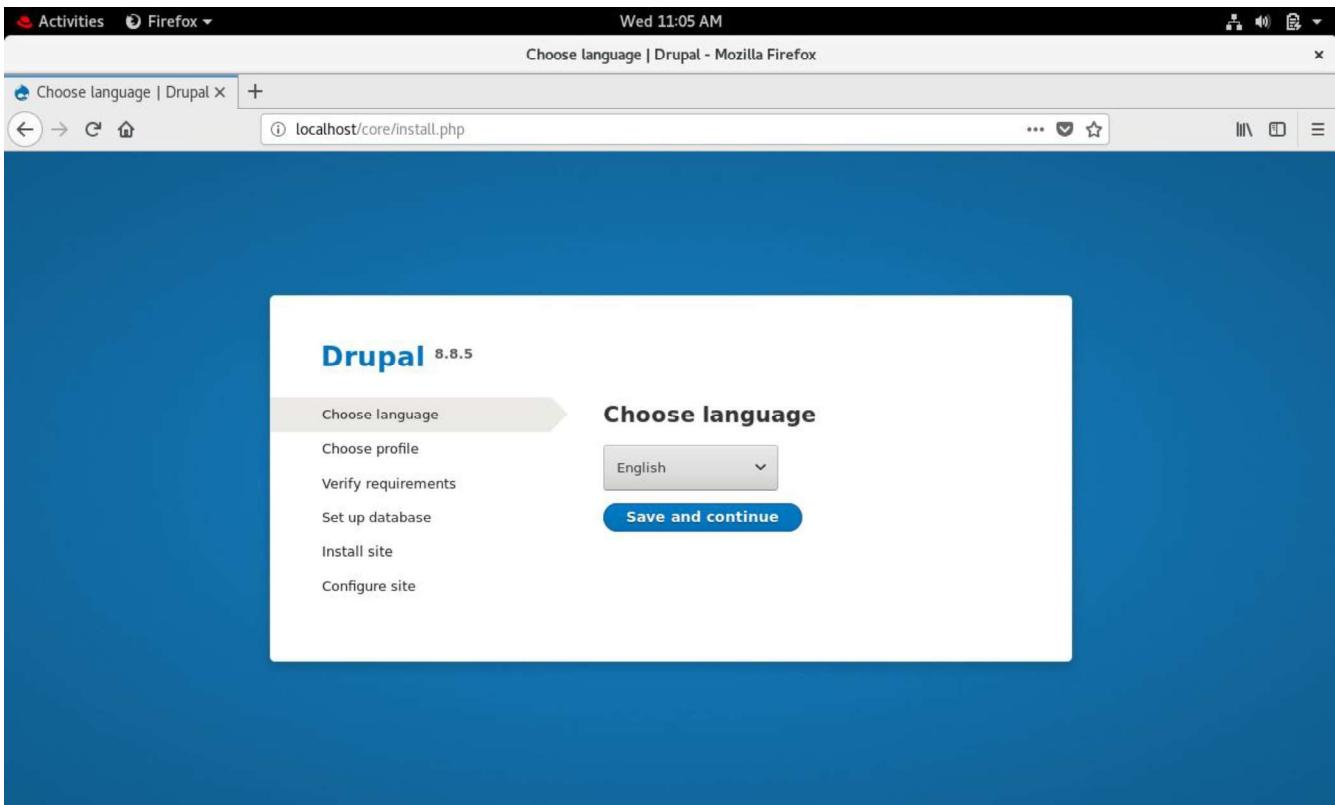
The screenshot shows a terminal window titled "Activities Terminal". The session is run as root on "localhost" at 11:44. The command history includes creating a directory (~/.my_drupal), listing files, navigating to the directory, opening a compose file with gedit, and running docker-compose up -d. The output of docker-compose shows the creation of several volumes and the pulling of the drupal:latest image.

```
[root@localhost ~]# mkdir ~/my_drupal/
[root@localhost ~]# ls
anaconda-ks.cfg  Desktop  Documents  Downloads  initial-setup-ks.cfg  Music  my_drupal  Pic
[root@localhost ~]# cd ~/my_drupal/
[root@localhost my_drupal]# gedit docker-compose.yml
[root@localhost my_drupal]# docker-compose up -d
Creating network "my_drupal_default" with the default driver
Creating volume "my_drupal_drupal_modules" with default driver
Creating volume "my_drupal_drupal_profiles" with default driver
Creating volume "my_drupal_drupal_themes" with default driver
Creating volume "my_drupal_drupal_sites" with default driver
Creating volume "my_drupal_db_data" with default driver
Pulling drupal (drupal:latest)...
[...]
```

Type following in docker-compose.yml for drupal... note that each compose files are in different folders...

```
version: '3.3'
services:
  drupal:
    image: drupal:latest
    ports:
      - 80:80
    volumes:
      - drupal_modules:/var/www/html/modules
      - drupal_profiles:/var/www/html/profiles
      - drupal_themes:/var/www/html/themes
      - drupal_sites:/var/www/html/sites
    restart: always
  postgres:
    image: postgres:10
    environment:
      POSTGRES_PASSWORD: your_postgres_password
    volumes:
      - db_data:/var/lib/postgresql/data
    restart: always
volumes:
  drupal_modules:
  drupal_profiles:
  drupal_themes:
  drupal_sites:
  db_data:
```

Once above commands are done type – **localhost:80** in your browser...



Enter same password you entered(**your_postgres_password**).. then it will install in few minutes...

The screenshot shows the 'Database configuration' step of the Drupal 8.8.5 installation process. The left sidebar lists steps: Choose language, Choose profile, Verify requirements, Set up database (which is selected), Install site, and Configure site. The main area is titled 'Database configuration'. It asks for the 'Database type*' and provides three options: MySQL, MariaDB, Percona Server, or equivalent; PostgreSQL (which is selected); and SQLite. Below that, it asks for 'Database name*', which is set to 'postgres'. It also asks for 'Database username*' and 'Database password', both of which are set to 'postgres'. A section titled 'ADVANCED OPTIONS' contains fields for 'Host*' (set to 'postgres'), 'Port number' (set to '5432'), and 'Table name prefix'. A note at the bottom of this section states: 'If more than one application will be sharing this database, consider using a different port number.'

Fill any information.. any password...

The screenshot shows the 'Configure site' step of the Drupal 8.8.5 installation process. The left sidebar lists steps: Verify requirements, Set up database, Install site, and Configure site (which is selected). The main area is titled 'Configure site'. It asks for 'Site name*' (set to 'Stock Market Prediction') and 'Site email address*' (set to 'devratneve@gmail.com'). A note below the email address field states: 'Automated emails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these emails from being flagged as spam.' Below that is a 'SITE MAINTENANCE ACCOUNT' section. It asks for 'Username*' (set to 'dewrat'), which includes a note: 'Several special characters are allowed, including space, period (.), hyphen (-), apostrophe ('), underscore (_), and the @ sign.' It also asks for 'Password*' and 'Confirm password*', both of which are masked. A note below the password fields states: 'Passwords match: yes'. At the bottom, there is a box with recommendations to make the password stronger: 'Recommendations to make your password stronger:' followed by two bullet points: '• Make it at least 12 characters' and '• Add uppercase letters'.

Sat 13:09

Configure site | Drupal - Mozilla Firefox

Red Hat | links - devwratneve@gmail.com | Install Drupal with Docker | ./install.sh => ERROR: Failed | Configure site | Drupal

localhost/core/install.php?rewrite=ok&langcode=en&profile=standard

Passwords match: yes

Recommendations to make your password stronger:

- Make it at least 12 characters
- Add uppercase letters

Email address *

devwratneve@gmail.com

REGIONAL SETTINGS

Default country

India

Default time zone

Kolkata

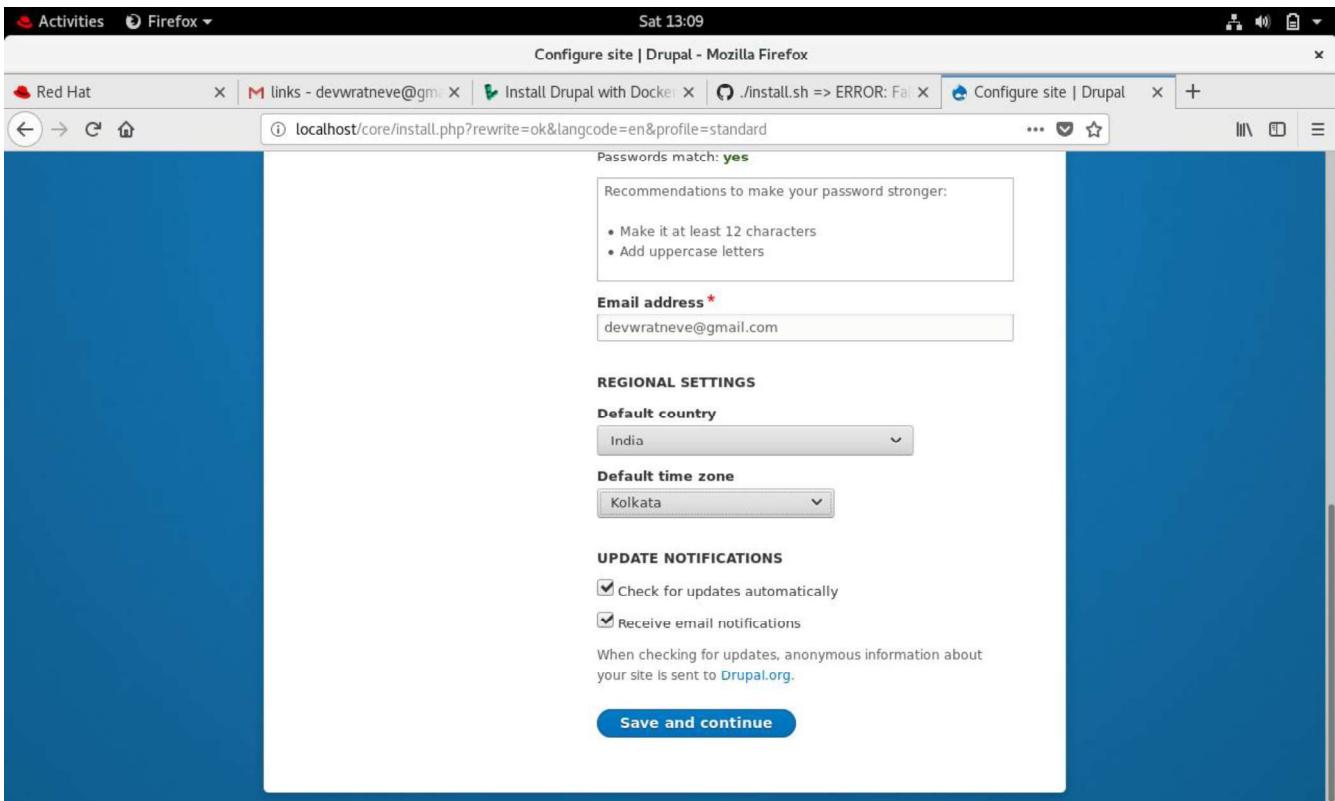
UPDATE NOTIFICATIONS

Check for updates automatically

Receive email notifications

When checking for updates, anonymous information about your site is sent to [Drupal.org](#).

Save and continue



And the website comes up... play with it.. write article.. you can yourself try whatever we did on owncloud...

Sat 13:10

Welcome to Stock Market Prediction | Stock Market Prediction - Mozilla Firefox

Welcome to Stock Market | localhost

Manage Shortcuts devwrat Edit

Content Structure Appearance Extend Configuration People Reports Help My account Log out

Stock Market Prediction

Home

Congratulations, you installed Drupal!

Search

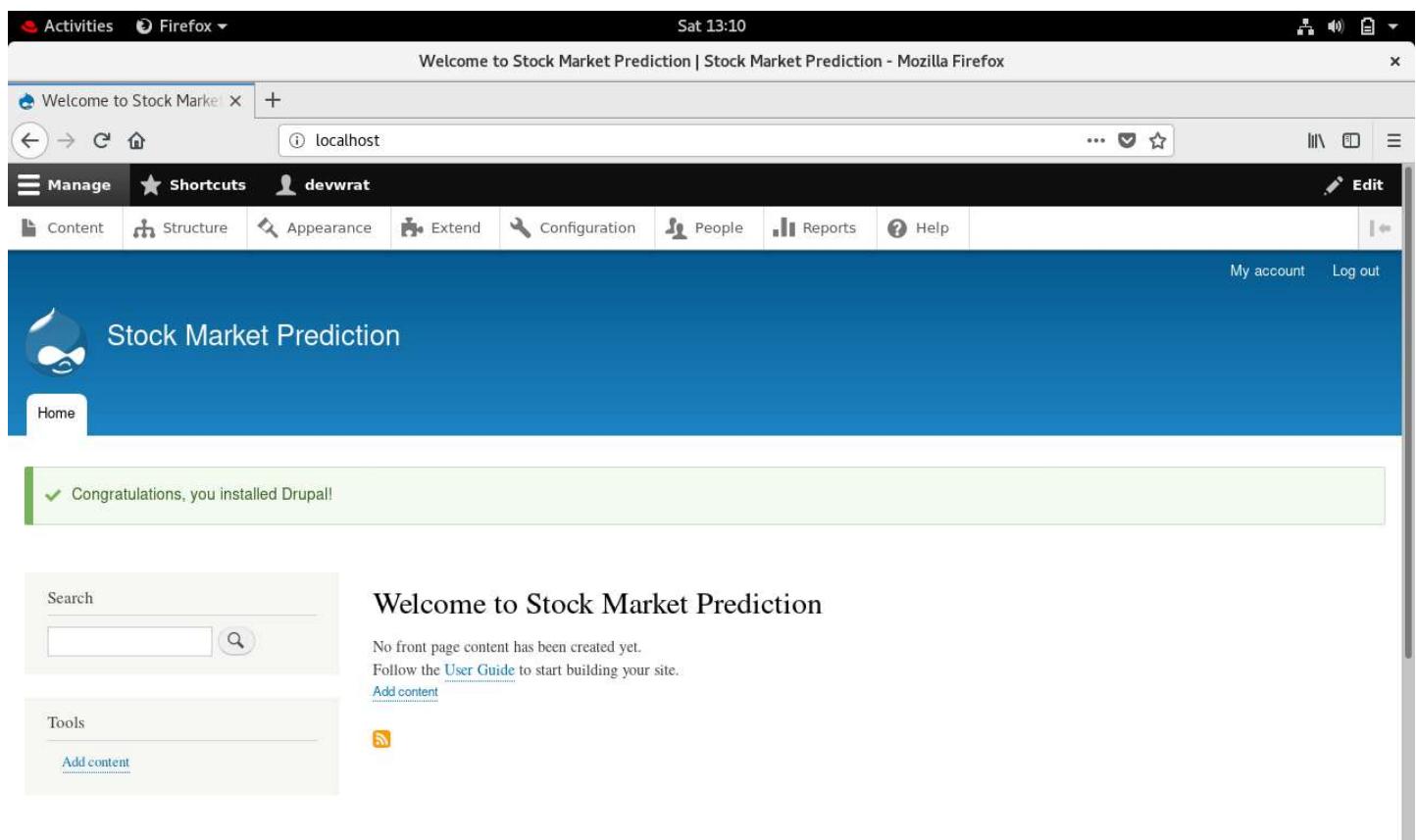
Tools

Add content

Welcome to Stock Market Prediction

No front page content has been created yet.
Follow the [User Guide](#) to start building your site.

Add content



You can also add users....

The screenshot shows a web-based administration interface for managing users. The top navigation bar includes links for 'Activities', 'Firefox', 'People | Stock Market Prediction - Mozilla Firefox', and a user icon 'devwrat'. The left sidebar has a 'Content' section with 'Structure', 'Appearance', 'Extend', and 'Configuration' dropdowns, and a 'People' section with 'Reports' and 'Help' dropdowns. The main content area is titled 'List' and shows a table of users. The table columns are 'USERNAME', 'STATUS', 'ROLES', 'MEMBER FOR', 'LAST ACCESS', and 'OPERATIONS'. One row is visible for 'devwrat', which is listed as 'Active' with the role 'Administrator', joined '26 minutes 23 seconds' ago, and last accessed '1 minute 46 seconds ago'. There is an 'Edit' button next to the row.

I wrote one small blog...

The screenshot shows a blog post titled 'About my BE project'. The post was submitted by 'devwrat' on Saturday, 05/02/2020 at 13:24. It features a profile picture of a man and a chart showing stock market data. The text of the post reads: 'I am in a last year of my engineering. I have made amazing project on Stock Market Prediction which predicts best stocks on daily basis... it works really amazing'. Below the post, there are links for 'Read more', '1 comment', and 'Add new comment'. A sidebar on the left contains a search bar and a 'Tools' section with a link to 'Add content'.

Done